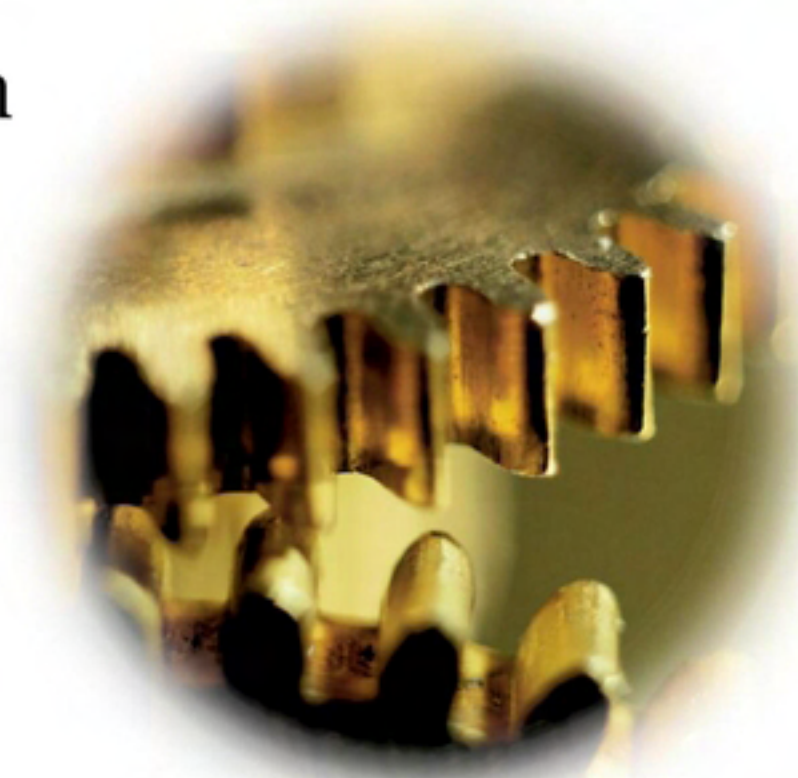




# Customizing and Upgrading Linux

Second Edition

Linda McKinnon  
Al McKinnon



# **Customizing and Upgrading Linux**





# **Customizing and Upgrading Linux**

---

**Second Edition**

Linda McKinnon  
Al McKinnon

**Gearhead Press™**




**Wiley Computer Publishing**



**John Wiley & Sons, Inc.**

Publisher: Robert Ipsen  
Editor: Ben Ryan  
Consulting Editor: Donis Marshall  
Managing Editor: Angela Smith  
Text Design & Composition: D&G Limited, LLC

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where John Wiley & Sons, Inc., is aware of a claim, the product names appear in initial capital or ALL CAPITAL LETTERS. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

This book is printed on acid-free paper. 

Copyright © 2002 by Linda McKinnon and Al McKinnon. All rights reserved.

**Published by John Wiley & Sons, Inc., New York**

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4744. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, (212) 850-6011, fax (212) 850-6008, E-Mail: PERMREQ @ WILEY.COM.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in professional services. If professional advice or other expert assistance is required, the services of a competent professional person should be sought.

The Gearhead Press trademark is the exclusive property of Gearhead Group Corporation.

*Library of Congress Cataloging-in-Publication Data:*

ISBN: 0-471-20885-X

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

*TEAM LinG - Live, Informative, Non-cost and Genuine!*

## A Note from Gearhead Press

---

Gearhead Press is dedicated to publishing technical books for experienced Information Technology professionals—network engineers, developers, system administrators, and others—who need to update their skills, learn how to use technology more effectively, or simply want a quality reference to the latest technology.

Gearhead Press emerged from my experience with professional trainers of engineers and developers: people who truly understand first-hand the needs of working professionals. Gearhead Press authors are the crème de la crème of industry trainers, working at the companies that define the technology revolution. For this reason, Gearhead Press authors are regularly in the trenches with the developers and engineers who have changed the world through innovative products. Drawing from this experience in IT training, our books deliver superior technical content with a unique perspective that is based on real-world experience.

Now, as an imprint of John Wiley & Sons, Inc., Gearhead Press will continue to bring you, the reader, the level of quality that Wiley has delivered consistently for nearly 200 years.

Thank you,  
Donis Marshall  
Founder, Gearhead Press  
Consulting Editor, Wiley Computer Publishing



## **Gearhead Press Books in Print**

---

(For complete information about current and upcoming titles, go to [www.wiley.com/compbooks/](http://www.wiley.com/compbooks/))

### **Books in the Gearhead Press Point to Point Series**

*Migrating to Microsoft Exchange 2000*

by Stan Reimer

ISBN 0-471-06116-6

*Installing and Configuring Web Servers Using Apache*

by Melanie Hoag

ISBN 0-471-07155-2

*VoiceXML: 10 Projects to Voice Enable Your Website*

by Mark Miller

ISBN 0-471-20737-3

### **Books in the Gearhead Press In the Trenches Series**

*Windows 2000 Automated Deployment*

by Ted Malone and Rolly Perraux

ISBN 0-471-06114-X

*Robust Linux: Assuring High Availability*

by Iain Campbell

ISBN 0-471-07040-8

*Programming Directory Services for Windows 2000*

by Donis Marshall

ISBN 0-471-15216-1

*Programming ADO.NET*

by Richard Hundhausen and Steven Borg

ISBN 0-471-20187-1

*Designing .NET Web Services Using ADO.NET and XML*

by Richard Hundhausen and Steven Borg

ISBN 0-471-20186-3

*Wireless and Mobile Development for Microsoft .NET Platforms*

by Steve Juntunen

ISBN 0-471-21494-9



*Making Win32 Applications Mobile: Porting to Windows CE*

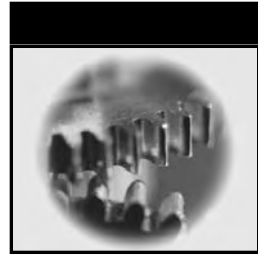
by Nancy Nicolaisen

ISBN 0-471-21618-6

*Programming Windows CE Wireless Applications*

by Barry Shilmover and Derek Ball

ISBN 0-471-21469-8



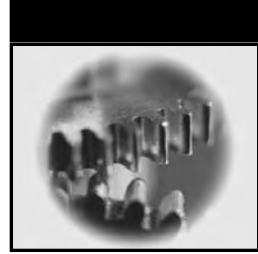
# Contents

<b>Acknowledgments</b>	<b>xiii</b>
<b>About the Authors</b>	<b>xv</b>
<b>Introduction</b>	<b>xvii</b>
<b>Chapter 1 Preparing for Installation: Chronicling Your Equipment</b>	<b>1</b>
Do Not Be Afraid; Just Do Your Homework	1
What Type of System Do You Have?	2
Discovering and Chronicling Your System's Equipment	4
<b>Chapter 2 Hard Disk Partition Planning and Preparation</b>	<b>7</b>
Swap Space Options	7
How Much Swap Space Is Required?	9
Hard Disk Partitioning	11
Master Boot Record and the Partition Table	14
The Boot Process	14
Linux and LILO	16

	Disk Partitioning: A Simple Approach	21
	Disk Partitioning: A Complex Approach	26
<b>Chapter 3</b>	<b>Installing Linux from a CD-ROM</b>	<b>35</b>
	Initiating a Linux Installation	36
	Creating the Install Boot Disk from a DOS/Windows System	37
	Creating the Install Boot Disk from a UNIX System	38
	Invoking the Linux Installation Program	40
	Entering Information in the Installation Program	41
	Installing Linux on an IDE System	42
	Stepping through the Installation	44
	Installing Linux on a SCSI System	68
	Reconfiguring the System	101
<b>Chapter 4</b>	<b>Installing Linux by Using NFS</b>	<b>103</b>
	Quick Review of Configuration	104
	NFS Configuration Details	105
	Preparing the NFS Server	105
	Preparing the NFS Client	109
	Typical NFS File Sharing	112
	NFS Solution for Windows	119
	Set Up the NFS Server	120
	Set Up the NFS Client	121
<b>Chapter 5</b>	<b>Configuring the Video</b>	<b>125</b>
	Video Signal Generation	125
	Video-Adapter Memory	126
	Basic Video-Adapter Bus	128
	Video Display	132
	Basic Monitor Design	132
	Video Frame and Synchronization	134
	How HSPs and VSPs Are Related	136
	Monitor Performance and Capabilities	139
	Monitor Age	141
	Bandwidth	141
	Vertical Frequency	144

Horizontal Frequency	144
Dot Pitch	144
Interlacing	147
Getting Ready for Video Configuration	149
The XF86Config File: Heart of the X Server	150
Using SuperProbe to Obtain Video Information	151
Using Xconfigurator to Create the Basic XF86Config File	153
Example Configuration	154
How to Install X from Scratch	167
<b>Chapter 6    Configuring Printers</b>	<b>173</b>
Basic Commands and Utilities	173
Configuring Local Printing	177
Technical Printing Details	185
Configuring Remote Printing	186
Prerequisites	186
<b>Chapter 7    Adding, Replacing, and Upgrading RPM Packages</b>	<b>195</b>
RPM Definition and Package Management	196
RPM Naming Convention	197
SRPM versus RPM	197
Determining A Package's Function and Contents	197
Applying Maintenance or Updating a Newly Installed System	200
Replacing and Upgrading the XFree86 Package	205
Using Packages to Install XF86Setup	209
Tutorial	210
<b>Chapter 8    Upgrading the Kernel by Using RPM Packages</b>	<b>213</b>
Why Upgrade or Patch the Kernel?	214
A Typical Security Advisory	214
Suggested Upgrade Procedure for Red Hat V7.X	215
A Quick Guide for RedHat V6.X RPM Upgrade	224
<b>Chapter 9    Configuring the Kernel</b>	<b>229</b>
Linux Kernel Configuration Menu	230
Code Maturity Level Options	231
Processor Types and Features	232

Loadable Module Support	233
General Setup	235
Plug and Play Support Option	237
Block Devices	237
Networking Options	240
Network Device Support	242
SCSI Support	244
Filesystems	247
Save and Exit	247
<b>Chapter 10 Installing a New Kernel by Building a New Source Tree</b>	<b>249</b>
Modules	250
The Modules Concept: How They Work	250
Finding a New Kernel	256
Installing a New Kernel	256
<b>Index</b>	<b>275</b>



## Acknowledgments

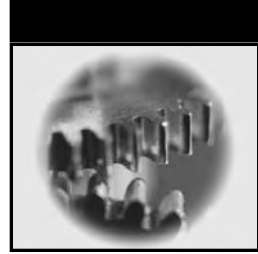
I have many to thank. First and foremost, thanks to Linda for the inspiration to write this book. Thanks to the many, many Linux developers all over the world. What a lesson in international cooperation we have all learned from you. Thanks also to all our colleagues and to all the students we have met over the past few years who expressed the need for a progressive introduction to Linux and also for a book they could refer to when they need it. We also thank the reviewers and readers of the first edition of this book, who made the thoughtful comments and suggestions that led to this “new and improved” version. And thanks to the editors at Gearhead and Wiley, whose questions and comments kept us on our toes and made us account for all the ideas, claims, and instructions found herein. Thanks to Miguel de Cervantes Saavedra (1547-1616) whose “Don Quixote of La Mancha” was the inspiration for the characters you will meet in this book. If you have not read “Don Quixote” and (especially) if your spirits need a lift, we wholeheartedly recommend it. Finally, as sentimental as it sounds, we still remember our beloved old dog Keeler, who spent her last few months watching us “pound out and mouse click” our way through the first edition. Her ghost lingers over this edition, too.

This book is dedicated to those who need or want a stable and robust operating system that is constantly and transparently undergoing improvement and to those Internet-linked free spirits who strive daily to make it even better.

—Al McKinnon

Thanks to all those who came to me and said, “Please write that down for us.” Without you, I would never have embarked on this literary tour. A special thanks goes to my peers who shared their views and to my students who shared their ideas and needs. During the past seven years, I have taught both basic and advanced Linux/UNIX curricula to more than 1,200 individuals. You know who you are; you guided my choice of topics. This book is dedicated to you and to those whom we have yet to convert from that other operating system.

—Linda McKinnon



## About the Authors

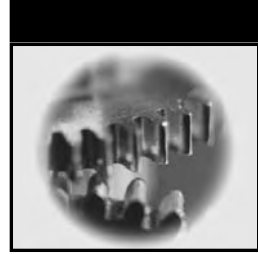
**Linda McKinnon** has more than 10 years of experience as a successful trainer and network engineer specializing in assisting major companies throughout the United States and Canada in network architecture, systems administration, and RFP procurement. In addition to being sought-after for her Linux expertise, McKinnon teaches large-scale server technology including the IBM RS/6000 SP and pSeries systems. On the software side, she teaches WebSphere Application Server Administration as well as WebSphere Commerce Suite implementation.

**Al McKinnon** is a professional trainer and engineer who assists clients throughout North America with network auditing, optimization and troubleshooting, and management and migration. McKinnon is also a technical author and editor for IBM and other clients, and he has extensive knowledge of networking technologies implemented on Linux, Windows 9x, Windows NT/2000, IBM AIX, Novell Netware, and IBM OS/2 Warp Server.

Al and Linda are both headquartered in Calgary, Canada, where the prairie meets the foothills of the Canadian Rockies.







# Introduction

While we were updating the book to reflect the more current Linux kernel and Linux distributions, we realized two things: 1) the older 6.2 distribution has become very pervasive, and 2) the newer 2.4 kernel is favored. Based on these discoveries, we decided not to remove much of the information regarding the 6.2 distribution and its default behavior. What we chose to do instead was add 7.1 distribution information. So now, in essence, this book covers both the 6.2 and 7.1 distributions, and it also demonstrates how the reader can upgrade either distribution to the most current 2.4 kernel.

We also enhanced our SCSI installation sections by replacing the SCSI hardware with RAID SCSI hardware. We performed this task for two reasons.

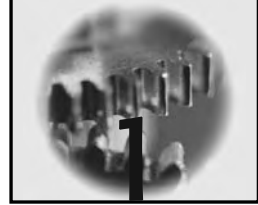
First, we wanted to add some sophistication to the book and show how easy it is to install Linux over a set of RAID drives. The system we used was a Compaq Proliant 800 series machine with a large external RAID tower (we also did not want to give the impression that our systems were typical, “in your basement” machines. This combination is typical of what the reader might see in any large data center).

Second, we wanted to further demonstrate how hard disk partitioning is practiced in the real world. The old cliché is true: A picture is worth 1,000 words, especially when it comes to working with partitions and mount points using RAID drivers.

There is an industry opinion now that Red Hat has become more sophisticated than the other Linux flavors because of all the IBM involvement. You can definitely see the influences, once they have been pointed out. For example, it was decided (by Red Hat) that for their 7.1 Server class installation, no X Window System components would be installed. This change is a big departure from the 6.2 Server class installation. Red Hat, of course, says that they implemented this policy because “most” large servers do not run X, so if you want it you have to install it (from scratch).

In the book, we made sure that the reader understands that this move was strategic and that they, the reader, have not screwed up their installation. Red Hat is moving toward a micro kernel, one that is modular in nature. This change required adding some information about how to handle modules. But while Red Hat has done a wonderful job of modularizing various components of its operating system, it has done its customers an injustice by not documenting this modular behavior. So, we tried to help by clarifying some points in our “upgrade” and “kernel compilation” procedures.

We also tried to complement Iain Campbell’s *Robust Linux: Assuring High Availability* book, also published by Wiley/Gearhead. Iain’s book nicely articulates in many areas exactly what we are talking about. For this reason, we think that if someone is going to start building and supporting Linux systems in a commercial environment (and not just for their own use), they should purchase his book *and* this book. Together, we cover most (if not all) of the issues that would be encountered during such installation and support.



# Preparing for Installation: Chronicling Your Equipment

Chronicling your equipment beforehand can prove to be very important to a Linux installation. If you have heard a horror story or two about Linux installations (overblown, in our opinion), chances are that the source of the horror was a lack of complete knowledge of the target system before proceeding. Everyone loves to shove that CD-ROM in and go.

In this chapter, you might be doing one or more of the following: using a hardware investigation utility, checking the system *basic input/output system* (BIOS), or even pulling the cover off the system to look inside. It all will be in the name of knowing what you are doing and moving ahead in a deliberate manner with everything under control.

## **Do Not Be Afraid; Just Do Your Homework**

---

Linux installation is not a horror story; rather, it is just a different kind of operating system installation. Any horror stories you might have heard are just myths and generally originate from one of the following three circumstances:

- The storytellers either did not investigate or did not chronicle their equipment for quick and easy reference prior to beginning their Linux installation.
- Perhaps their equipment was too new or too old. Their version of the Linux kernel might not yet have had the necessary drivers or support for that particular new hard disk, video card, and the like.
- Older BIOSs that do not support the new Plug-and-Play devices.

The *trick of the trade* is to avoid spending too much time with a system that exhibits unpredictable behavior before reaching for a suitable utility to investigate the situation—especially if you have just taken the opportunity to add or change a device. The first thing you should check for is a potential misconfiguration. If the system promises Plug-and-Play support and is not functioning predictably, the BIOS level in your system might need updating. Bet you did not know this function is part of your BIOS and not the hardware. This function is not a Linux problem. This situation would confuse Windows 95, given the chance. Older BIOSs might not work properly with Plug-and-Play devices, and this system is the first thing you will need to upgrade or change or take matters of IRQ assignments into your own hands. The reason is that each BIOS manufacturer has its own algorithm for assigning the next available resource to a potential Plug-and-Play device. Unfortunately, we have to direct you to [www.hardwarebible.com/](http://www.hardwarebible.com/) at this time, because an in-depth conversation about hardware conflicts is outside the scope of this book.

Some of the most recent improvements made to the new Linux kernel version 2.4 have to do with increased feature and driver device support for both new Plug-and-Play as well as larger systems so that fewer of these experiences occur. We are going to present two installations. One uses an *Integrated Drive Electronics* (IDE) drive configuration to represent a typical desktop system, and another uses a SCSI RAID subsystem to illustrate how easy it is now to build a large server using Linux as an operating system. We will carefully chronicle all of the equipment used in each case. These installation notes can be used as a model for your own Linux implementation.

## **What Type of System Do You Have?**

You might be building a brand new system or trying to install Linux on an older system. We will try to draw your attention to, and comment on, all of the issues that pertain to both scenarios.

### ***Using Recently Purchased Equipment***

If you are installing on recently purchased equipment, you must take the time to do some research on the Internet to learn which systems are or are not supported by Linux. Yes, some play better than others do. It is usually not so much the system itself as the network interface or video cards inside them. There are some very good Web sites that specialize in this information, including the “gritty stuff” on hard drives, controllers, modems, network cards, sound cards, ports, memory, and video chips, including the monitor. The definite hardware reference site is [www.redhat.com](http://www.redhat.com). From there, you can follow links to the hard compatibility lists. Be meticulous.

You should be obsessed with ensuring that you receive from vendors all technical booklets on all parts within the system. At the risk of sounding redundant, this information includes all detailed information on the hard drive, controllers, modems, network cards, sound cards, ports, memory, and video chips, including the monitor. We cannot emphasize enough the importance of knowing the exact details of the video and monitoring technical specifications. If a dealer is assembling your system, make sure that you get all the booklets and utility disks that shipped with every card inside your new system before you leave the premises.

The “bench guys” that work on these systems collect utilities in case they ever have to solve an issue with a customer’s system. The utilities could be yours. If at all possible, you should try to deal with a company that is willing to give you copies of any new drivers or utilities that it has had to acquire in order to correctly set up your system. The drivers and utilities that ship with the interfaces are often obsolete, so the “bench guys” have to acquire and use their own. Needless to say, the dealers and technicians that share this information are worth their weight in gold.

### ***Using Existing Equipment***

Regarding installation on older or existing equipment, generally the more established and better supported the system, the better off you will be. There are some exceptions, too, however. We recommend that you perform the same checks on hardware, drivers, and utilities that you would make on any new system. You might end up replacing unsupported hardware. Because of the new interest in Linux, valuable Web sites have emerged that specialize in assisting you with acquiring documentation on systems, interfaces, and adapters. Again, their premier reference site is [www.redhat.com](http://www.redhat.com). You might have to take the cover off the system and start identifying the various adapters and video cards right down to the

chip sets. Information on monitors can be tough to come by, in which case you might have to rely upon Linux support groups that have taken the time to share their experiences. We must also give you some advice here. If you have invested a lot of time looking for information on a specific adapter and just cannot find anything, then it probably is not supported. Get another adapter and move on.

In general, Linux will run on a 386 or 486 with 16MB of memory. Do not expect to get a decent X Window System desktop, such as KDE (the K Desktop Environment) or GNOME, to perform decently with only 16MB of memory. It is our experience that you will require at least 32MB of memory to run an X Window System desktop. Having said that, a system with 16MB of memory will work very well at the command line if that is sufficient for your needs.

## **Discovering and Chronicling Your System's Equipment**

There are many good tools available both on the retail market and via the Internet to help you discover what is inside your system. The trick is to find one that does not require the installation of another operating system first. Ideally, you should find a utility that works on a system that has just been booted by a single bootable *disk operating system* (DOS) diskette. It seems silly and unproductive to have to load another operating system on a machine just to be able to run probing utilities. We have been in the industry for years and have always kept a set of DOS boot disks. Nothing is so uncomplicated and beautiful than booting a system with DOS and probing hardware for possible IRQ conflicts and chip set information. For those readers who are pre-DOS, you can go to a site called [www.theusefulsite.com/dos\\_utils.htm](http://www.theusefulsite.com/dos_utils.htm) or [www.dc.ee/Files/Utils](http://www.dc.ee/Files/Utils) to see what is available.

We are providing an example component list as follows (see Table 1.1). This list was taken using the DOS utility called Hwinfo, which was downloaded from [www.dc.ee/Files/Utils](http://www.dc.ee/Files/Utils).

It is very important to capture at least as much detail on the system as possible. The installation process will ask you to make choices and decisions based on the type of devices that you have inside your system, and you need to be ready to respond correctly. Guessing at these responses creates those horror stories that you hear about. This task is not as daunting a job as one would think, because most utilities including the one we chose to use have the capability to save the results to a file that you can print later for documentation. At the bottom of the table are some networking details

**Table 1.1 Components of the Example System**

COMPONENTS	NAMES/SPECIFICATIONS/SETTINGS
Main Processor	Intel Pentium Pro 1GHZ
Math Co-Processor	Internal, 9,527.0 MHz
Bus Type	ISA + PCI Local Bus
Memory	768MB
BIOS Manufacturer	American Megatrends
Network Ethernet Adapter	RealTek Semiconductor 8029 PCI
Video Adapter	VGA
Monitor Type	VGA Analog color
Video Chipset	S3 unknown
Video RAMDAC Type	Standard VGA
Video BIOS Manufacturer	S3 Inc.
Video BIOS Version	2.01.16J
VESA Support Present	V2.00
VESA Manufacturer	SC Incorporated. ViRGE /DX /GX
Fixed Disk Controller	Enhance IDE
Fixed Disk Capacity	2,389 Mbytes
Fixed Disk Physical Geometry	Cylinders: 13,176, Heads: 16, Sectors/Track: 63
CDROM	Yes
Mouse type/no. of buttons	PS/2, 2 buttons
Networking Details	Host Name: HostA IP: 192.168.1.206 Mask: 255.255.255.0 Default Gateway: 192.168.1.1 DNS Server: 24.64.2.33 DNS Server: 24.64.2.34
Printer	LPT1: HP LaserJet 6



that will be required to connect the system to the network. If you do not manage the network, then the person who does will have to supply you with this information. If you are on a commercial network, please do not just choose any IP address or you will create havoc. Make sure that the IP address is valid and available before you use it. If you have and are not sure, then simply remove the cable from the back of the Ethernet adapter so you do not appear as a new host on the network until you are ready.

The next chapter covers best industry practices for partitioning the hard drive as well as a typical installation.



# Hard Disk Partition Planning and Preparation

Two major issues face users who are new to the installation and configuration of Linux platforms. One of them is how to handle hard disk partitioning and the other is dual booting.

Linux users encounter the disk partitioning issue almost immediately upon beginning an installation. The first topic we address in this chapter is swap space options, because the decision to create a *swap partition* versus a *swap file* must be made first. Both options work, but one might be more appropriate. This decision will depend on your performance expectations. We discuss both in the following section.

After covering swap space options, we turn to dual booting and hard disk partitioning. We then illustrate the roles of partitions and other software during the boot process. Finally, we present examples of simple and complex disk partitioning strategies and disk usage.

## Swap Space Options

---

Before we begin this discussion, it is useful to note that swap space is also referred to as *disk paging*, and we will not differentiate between the two. The

concept is still the same, however, whereby Linux will write *pages* of memory to the hard disk when *Random Access Memory* (RAM) becomes either overextended or old. It is the responsibility of the operating system to clean up or free RAM. The point we are trying to make is that you will never be able to eliminate paging. Paging is inevitable and healthy under the right circumstances. What you do want to do is make sure that there is sufficient RAM inside the system to avoid paging out information prematurely due to a lack of RAM. Otherwise, the system *thrashes*. *Thrashing* occurs when an application pages information prematurely out of RAM to free up RAM to read in more information required for the current operation. In other words, before the current operation is finished, it has to retrieve that paged information back from the swap partition on the hard drive back into RAM to continue. Meanwhile, you are wasting *central processing unit* (CPU) cycling information between the hard disk, RAM, and CPU, rather than just the RAM to the CPU. This situation is your worst performance nightmare. Avoid this situation at all costs in a production system. In other words, configuring swap space is not an acceptable fix for a lack of system RAM. The amount of RAM required in a system depends entirely upon how many applications exist and the nature of the applications that are going to be running on that system. Think of swap space as a working area for the operating system to page out stale information that is not immediately required in RAM by the application that is currently running and not as a substitute for physical RAM.

Swap space can be a dedicated partition, called the swap partition, or alternatively a swap file. The swap file means what its name implies. It will be no more than a file within the installed directory structure. There are pros and cons in the use of each option. The issues are twofold: first, which option do you need or want; and second, how large should you make the partition(s) or file(s)?

We always recommend a dedicated swap partition because you will get better performance. Because the swap partition is created before the installation, you have to preplan so that you can create the necessary partition types and sizes during the installation. If you install Linux using all of the defaults, Linux will create a swap partition and not a swap file. The swap partition will ultimately show up as an extended partition within the primary Linux operating system partition. The following “Fdisk Output” example shows the results of having installed Linux inside one single partition with a swap partition, then viewing them with `fdisk`. The `/dev/hda1` is the single partition that was created with a mount point `/` signifying the root partition and is a primary partition. A *mount point* indicates to the directory that the partition will be mounted. Once a partition is mounted, the partition can be

accessed and used. The `/dev/hda2` is the extended partition and does nothing except act as a container for the `/dev/hda5`. We can see this function because `/dev/hda2` and `/dev/hda5` both have the same start and end sector numbers. Also, note that the swap partition does not have a mount point reference. It is a logical device that is entirely system managed.

### Fdisk Output

```
Disk /dev/hda: 255 heads, 63 sectors, 2482 cylinders
Units = cylinders of 16065 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
<code>/dev/hda1</code>	*	1	2418	19422553+	83	Linux
<code>/dev/hda2</code>		2419	2482	514080	5	Extended
<code>/dev/hda5</code>		2419	2482	514048+	82	Linux swap

You should not entirely rule out the use of swap files, however. Swap files can be created after the installation in response to an emergency or temporary situation. They will work in conjunction with the rest of your computer memory configuration. In other words, swap files can be added or removed without having to repartition. Other than creating the swap file, there is nothing special you have to do to get the swap file to participate in the paging process (which already includes your existing swap partition). The Linux operating system will automatically use both. It is best practice to deactivate and remove the swap file(s) as soon as your situation is resolved. If long-term solutions are required, consider adding RAM or repartitioning the hard disk. Do not rely on using the comparatively inefficient swap files for anything beside the short term. Because a swap file can only be added to a system that is already running, it is deemed to be a system administration task.

## How Much Swap Space Is Required?

If your system has only 16MB of RAM or less, a swap partition is mandatory. The system will utilize this space as a working area to save and merge files during the booting phase. The swap partition has to be at least 16MB. If you are going to use the X Window System or any of the desktops, you will probably need at least 32MB; otherwise, performance will be quite unacceptable.

The amount of swap space you choose to create will ultimately depend on what kinds of application software will be running on the system. A reasonable, albeit rough, estimate is a size equal to twice the amount of RAM. For example, if you have 32MB of RAM, then you would configure 64MB of swap space. This situation would hold true up to about 512MB of RAM,

and then you would have to exercise some discretion. For example, if the system has 512MB of RAM and is only running a single e-mail or *Simple Mail Transport Protocol* (SMTP) application for fewer than 900 accounts, then you would probably only require 776MB of swap space. If the same system were destined to be a database server-Web server combination, however, then you would probably configure 1GB of swap space. The following two examples show statistics from both the `/proc/swaps` and `/proc/meminfo` files, respectively, for two different hosts to demonstrate what we mean. We accepted all of the defaults during the installation to demonstrate how Linux will automatically calculate how much swap space is required to support various amounts of RAM. HostA has 789553152 bytes or 790MB of RAM, and Linux created a 1612111872 byte or 1.6GB swap space. HostB has 524574720 bytes or 524MB RAM, and Linux created a 1077469184 byte or 1.1GB swap space. This relationship is almost two to one.

### HostA RAM and Swap Details

```
[root@HostA /root]# cat /proc/swaps
Filename                                Type              Size      Used      Priority
/dev/hda5                              partition         1574328  0         -1

[root@HostA /root]# cat /proc/meminfo
          total:      used:        free:  shared: buffers:  cached:
Mem:   789553152 38703104 750850048          0  2084864 16551936
Swap: 1612111872          0 1612111872
MemTotal:        771048 kB
MemFree:         733252 kB
MemShared:           0 kB
Buffers:          2036 kB
Cached:           16164 kB
Active:           18116 kB
Inact_dirty:        84 kB
Inact_clean:         0 kB
Inact_target:       480 kB
HighTotal:          0 kB
HighFree:           0 kB
LowTotal:          771048 kB
LowFree:           733252 kB
SwapTotal:        1574328 kB
SwapFree:         1574328 kB
```

### HostB RAM and Swap Details

```
[root@HostB /root]# cat /proc/swaps
Filename                                Type              Size      Used      Priority
/dev/hda6                              partition         1052216  0         -1

[root@HostB /root]# cat /proc/meminfo
          total:      used:        free:  shared: buffers:  cached:
Mem:   524574720 221384704 303190016          0  29192192 88784896
```

```

Swap: 1077469184      0 1077469184
MemTotal:      512280 kB
MemFree:       296084 kB
MemShared:      0 kB
Buffers:       28508 kB
Cached:        86704 kB
Active:        113164 kB
Inact_dirty:    2048 kB
Inact_clean:    0 kB
Inact_target:   3328 kB
HighTotal:      0 kB
HighFree:       0 kB
LowTotal:      512280 kB
LowFree:       296084 kB
SwapTotal:     1052216 kB
SwapFree:      1052216 kB

```

Large systems with 2GB of RAM and higher generally have an entire drive and controller dedicated for swap space so that the operating system reads and writes for swap space information will not compete with production data reads and writes. It would be considered bad practice to configure a swap space on a very busy data drive. Instead, choose a drive that has very little or no other activity. Better yet, dedicate an entire drive to the swap space alone.

## Hard Disk Partitioning

---

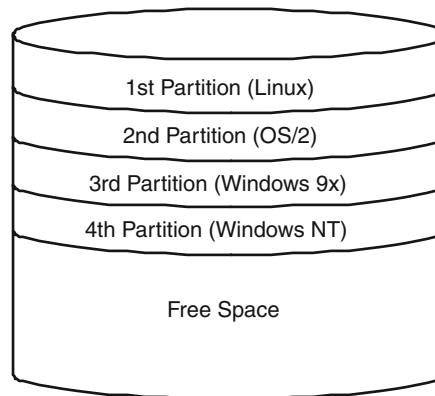
The hard disk is, of course, the primary medium for storing information. Hard disks grow in size every year, and as they get larger, using them in an efficient way becomes more difficult. The file system is the general term used to describe the logical structures and software routines used to control the access to the storage on a hard disk system. Different operating systems use different ways of organizing and controlling access to data on the hard disk, and this choice is independent of the specific hardware used. The information in this section, in fact, straddles the fine line between hardware and software—a line that gets increasingly blurry every year. The discussion that follows starts with explaining the logical structures and ends with physical hard disks.

The nature of the logical structures on the hard disk has an important influence on the performance, reliability, expandability, and compatibility of your storage subsystem. This section looks at how these logical structures are set up and used. Of particular concern is how they relate back to the physical drives.

The three basic types of partitions are primary, extended, and logical. You can create only four primary partitions on a single physical drive, and only one primary partition can be an extended partition. An extended partition can hold a maximum of 12 logical partitions, however. Bill Gates started it, and this situation is how it ended up.

This situation creates an environment where we can take a single physical drive and put a different operating system in each primary partition, in which case you might have a management scenario as that shown in Figure 2.1. If your Linux system is going to a production system, however, we do not recommend placing different operating systems on the hard disk. If you are just installing Linux on a test system or at home, however, you might wish to install other operating systems for many reasons.

If you are just starting out and you do not wish to use the standard Linux default partitioning, we recommend that you use Disk Druid. The Disk Druid tool will walk you through the creation of partitions. If you are somewhat experienced with partitioning, you might want to use fdisk or any other tool of your choice. Fdisk will give you more flexibility at the expense of experience. Experts can download a tool called the *First Nondestructive Interactive Partition Splitting* (FIPS) program, which will enable you to manipulate your existing DOS partitions without damaging your data (providing that you use the tool correctly). A product called Partition Magic is also popular for disk repartitioning and is not restricted to DOS partitions. The issue at hand is always the same. Reducing partition sizes is tricky and does not guarantee that you will not lose data. You can increase the size of a partition providing you had the foresight to leave some unused space on the



**Figure 2.1** Hard disk partitioning.

drive. If you failed to leave free space, you can always add a new drive. What is very difficult is reducing the size of a partition once it has data in it. This situation often requires backing up the data, recreating the partition, and then restoring the data to the new partition. This procedure works if the partition holds application data, because you can stop the application, deal with the data, and then restart the application. When you are dealing with the partition that the operating system is using, however, it is a different story. You cannot stop the operating system to reduce the size of the partition and then effectively restore it without jumping through firing hoops. Planning partition sizes is a very important step.

The four tools mentioned earlier are not the same. They have very different default behaviors. For example, if you use `fdisk` to create a set of multiple file system partitions for Linux, you have to specifically create a primary partition or extended/logical partitions. In contrast, if you create the file system partitions by using the Disk Druid tool, Linux automatically organizes each partition you create into logical partitions for you and maps them to a mount point.

Be careful when placing different partitions for different operating systems on a single hard disk. For best results, if you wish to create four partitions containing DOS, Windows 9x, OS/2, and Linux, use each operating system's respective disk-partitioning tool to create the partition for that operating system. For best results, use the `fdisk` (or equivalent) tool that shipped with the product. This usage will ensure compatibility between the partition and the operating system. The exception here might be the OS/2 version of `fdisk` created by IBM or by a professional partitioning tool. You can use it to create *all* partitions, but then you might have to go back to Linux `fdisk` to change the partition type to native Linux anyway.

When starting with a blank disk, you should always begin with the least-sophisticated partition manager first and work your way up. The customary sequence is as follows: Windows 9x, Windows NT/2000, OS/2, and then Linux `fdisk` or Disk Druid. We should also mention that Red Hat's service agreement has limited support for running multiple operating systems. The Red Hat agreement actually reads, "Multi-boot installations with three or more Oss, Don't even think about it—you're on your own. :)" This excerpt is from [www.redhat.com/support/sla/SLAdescriptions.html](http://www.redhat.com/support/sla/SLAdescriptions.html).

At the end of the partitioning process, you will have created the "ultimate partition table." A partition table is just a map of all the drive partitions and where they start and end on the drive. The partition table resides at sector 1 of cylinder 0 of the physical hard disk. With respect to the physical disk, the partition table resides at the same location along with the



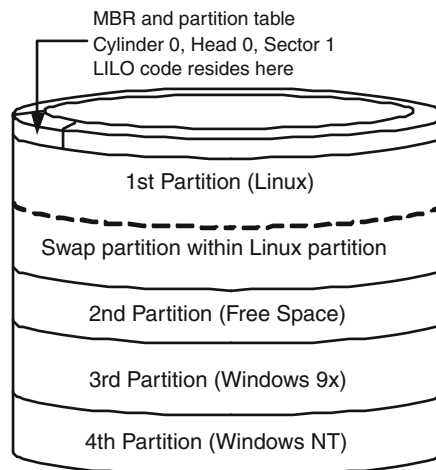
*Master Boot Record* (MBR). The MBR is the “tricky code and magic” that sits at sector 1 as well that will enable you to eloquently configure which partition you would like to boot from in the form of a menu. This code is called the *LinuxLoader* (LILO). In OS/2, this code is called the Boot Manager. Many products in the marketplace are good. We are going to focus on LILO, however. Either will work, but we are going to show you how to configure LILO in the next section after we explain more about the relationship between the MBR and the partition table.

## Master Boot Record and the Partition Table

The MBR is always on the first sector of the physical hard disk, as shown in Figure 2.2. That first sector is numbered or called, somewhat idiosyncratically, “cylinder 0, head 0, sector 1.” The partition table is always located alongside the MBR. For this reason, they often refer to this area as the “partition boot record.”

## The Boot Process

Our intent in this section is to explain how the MBR and the partition table work together to contribute to booting the system. Most PC architectures behave in the same manner that we are going to describe. The process is fairly generic; that is, regardless of the operating system installed, the process is



**Figure 2.2** Master boot record and the partition table.

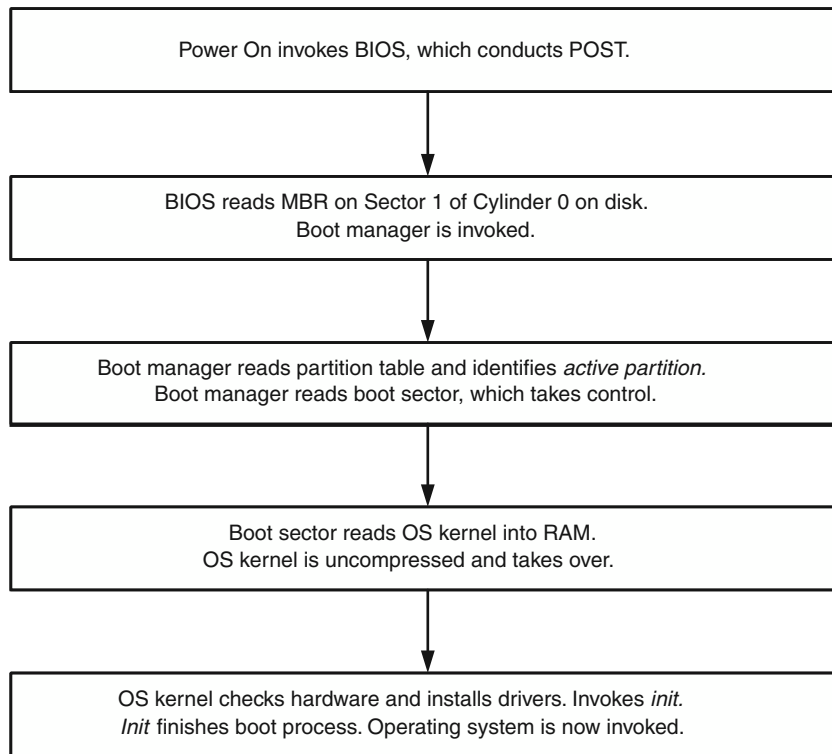
much the same. Only the last few steps of the process, involving both the root file system and `init` process, can be considered unique to Linux.

When a PC is turned on, a small boot program that resides on the PC's *Read-Only Memory* (ROM) BIOS chip performs a *Power On Self-Test* (POST). The purpose of the POST is to check the basic hardware and assign any Plug-and-Play parameters required by boot devices. Immediately after the POST, the boot program (also part of the BIOS) sets off in a quest for sector 1 with the hopes of finding where the operating system to copy to RAM is located.

When the PC's *Complementary Metal-Oxide Semiconductor* (CMOS) is configured such that the BIOS is to search a hard disk, then the BIOS will begin by searching sector 1 of the hard disk for further instructions. These instructions are in a small program called the boot manager or bootstrap loader. The BIOS will then load a copy of the MBR/boot manager into the RAM and then branch to it. *Branches* means to pass control from the BIOS to the boot manager so that the boot manager can take over. The boot manager within the copy of the MBR reads the partition table, which in turn points to the boot sector within the partition table that is called the *active partition*. Active partition is a holdover term from the DOS days and means bootable. The boot manager now has everything it needs to read the boot sector, and the boot sector takes over. The boot manager code contains the data that points to the physical device (disk) that points to the logical device (partition) that points to the file (kernel) that needs to be loaded in RAM.

The boot sector then reads the compressed operating system kernel into RAM, and a small program at the beginning of the kernel will uncompress the kernel. The kernel takes over and continues the boot process by first checking for other installed hardware and installing the appropriate drivers. Then, it starts mounting the root file system. The root file system has an *inittab* file that invokes the `init` thread (pid 1) process, which in turn invokes the other processes and daemons depending on its configuration. The `init` process also switches to the default or specified run-level mode and invokes the `getty`, which is the code that generates the login prompt. `getty` also opens `tty` lines and sets their modes. At this point, the PC is finished booting and is ready for logging into. This process is summarized in Figure 2.3.

We admit that we have simplified the boot process, but in summary, the MBR generally houses the boot manager and the boot manager reads the partition table to find the location of each partition installed on the physical drive. If each partition contains a unique operating system, then each will also probably contain a boot sector and kernel for its operating system.



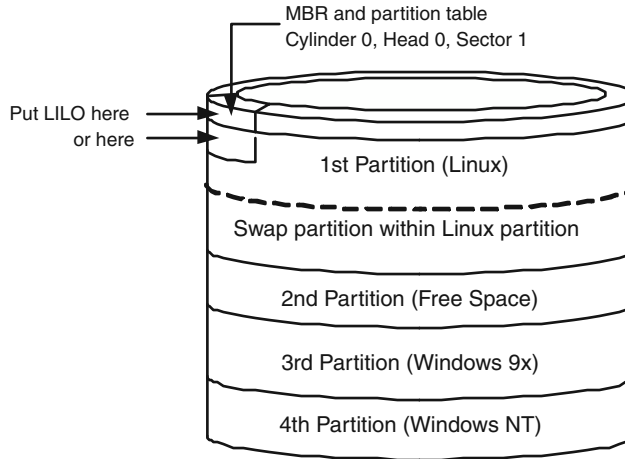
**Figure 2.3** The boot process.

## Linux and LILO

You have probably inferred from our discussions thus far that most operating systems boot off a primary partition, not a logical partition. In reality, Linux is an exception to this general rule. Linux does not require a primary partition because LILO can be configured to reside in a boot sector of an active partition block.

LILO is a small boot-manager program that can be configured to reside in one of three places: inside the MBR, on a floppy disk, or inside the boot sector of an active disk partition, as shown in Figure 2.4. Other available boot manager programs that do the same tasks as LILO are OS/2's Boot Manager and Partition Magic (which uses OS/2's Boot Manager). You should not use more than one of these programs, however, because they do not work together. If you contemplate running OS/2 on any of the partitions, we suggest that you use the OS/2 Boot Manager instead of LILO.

Thus, LILO is not your only option for a boot manager. You can do without it or use some other boot manager. On any system that does not multi-



**Figure 2.4** Linux and LILO, the Linux Loader.

boot with OS/2, however, we recommend using LILO. Thus, when you choose LILO to be the boot manager, it can be configured to boot Linux by default or enable you to choose (via a small simple procedure) which partition you wish to boot (meaning that you can choose which operating system you wish to use). Another decision you have to make in a multi-OS configuration is where to install LILO.

### ***Where to Install LILO***

It is best practice to install LILO in the MBR for a production server. We refer to a production server as a server that is going to provide a stable service on the network to users or other applications. The reason is that production servers are always planned with a solid recovery procedure in place, and Red Hat will not support any other boot loader than its own. Also, the MBR is a special area on the hard disk that is automatically loaded by the system's BIOS and is the earliest point at which LILO can take control of the boot process, which varies between BIOSs. You can easily tell whether a system has LILO installed in the MBR because you will be presented with a `boot :` prompt as opposed to a boot manager type of menu.

### **Rules of Thumb**

If you are going to build a dual boot system managed by LILO, then here are some rules of thumb. For systems that are running Windows 9x along with Linux, install LILO in the MBR so that LILO can boot both operating

systems. You will have to modify LILO's */etc/lilo.conf* after the Linux installation to point to the Windows 9x boot kernel.

For systems running Windows NT along with Linux, install LILO in the first sector of the root partition (and not in the MBR) and use a boot manager. The reason is that both Windows NT and Linux want to write information to the MBR. If you install Windows NT, first it will populate the MBR with its information. Then, Linux will also populate the MBR with its information and corrupt the MBR. This situation holds true for the alternate case, as well. They both are designed to use the MBR method for booting. If you insist on using the MBR, then you will either need to use boot diskettes or reconfigure the NT system loader to boot LILO from the first sector of the root partition. You know what? It is easier to use the alternate method of using LILO from the first sector of the root partition.

In the event that you are not going to use the MBR and you are going to utilize the first sector of the root partition, you will need a boot manager. To achieve this task, put LILO in the first sector of the root partition so that the boot manager can manage it. How is this process done? During the installation, you will be asked where you want to install LILO. Choose "First sector of the boot partition" and not MBR when presented with the LILO install menu.

In this case, your other boot manager will take control before LILO. You can then configure the boot manager to start LILO or any other operating systems that are on the same system. In summary, to boot Linux you will still have to invoke LILO, but LILO itself will not be the first in control—the boot manager will be.

### **Without LILO**

This question is simple. If you choose not to install LILO for any reason, you will not be able to boot Linux directly from the hard disk. You will have to use a boot diskette or boot over the network.

### **Using LILO**

If you have chosen to let LILO manage the disk, then you can access the boot sequence by using the Tab key precisely at the `boot :` prompt during initialization. If your timing is correct, a list of available operating systems, as configured in the */etc/lilo.conf* file, will appear.

When you install Linux, a */etc/lilo.conf* file will be built for you. Here is the */etc/lilo.conf* file from HostA in our examples:

```
[root@HostA /boot]# cat /etc/lilo.conf
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=linux
image=/boot/vmlinuz-2.4.2-2
    label=linux
    read-only
    root=/dev/hda8
```

This file can be interpreted as follows:

- `boot=/dev/hda` tells LILO to look on the first hard disk on the first IDE controller.
- `map=/boot/map` locates the map file. This file is a binary file created by the system and should not be modified.
- `install=/boot/boot.b` tells LILO to install the specified file as the new boot sector. `map=/boot/map` locates the map file. This file is a binary file created by the system and should not be modified.
- The existence of `prompt` tells LILO to show you whatever is referenced in the message line. While it is not recommended that you remove the prompt line, if you do remove it, you can still get a prompt by holding down the Shift key while your machine starts to boot.
- `timeout=50` sets the amount of time that LILO will wait for user input before proceeding with booting the default line entry. This time is measured in tenths of a second, with 50 as the default.
- `message=/boot/message` refers to the screen that LILO displays to enable you to select the operating system or kernel to boot.
- `linear` describes the hard disk geometry to LILO. Another common entry here is `lba32`. You should not change this line unless you are very aware of what you are doing. Otherwise, you could put your system in a state where it cannot boot.
- `default=linux` refers to the default operating system for LILO to boot from the options listed below this line. The name `linux` refers to the label line below in each of the boot options.

- `image=/boot/vmlinuz-2.4.2-2` specifies the linux kernel to boot with this particular boot option.
- `label=linux` names the operating system option in the LILO screen. In this case, it also is the name that is referred to by the default line.
- `read-only` specifies that the root partition (see the root line below) as one that cannot be changed, only read.
- `root=/dev/hda8` tells LILO what disk partition to use as the root partition.

The following */etc/lilo.conf* file was modified to boot both linux and DOS. We recommend that you make a copy to */etc/lilo.conf.bak* before you modify the original. The following sample */etc/lilo.conf* file creates two boot choices: linux or DOS. These choices will be keyed off the `label=linux` and `label=dos` statements in the file.

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt=
timeout=50
image=/boot/vmlinuz-2.4.2
    label=linux
    root=/dev/hda8
    read-only
other=/dev/hda1
    label=dos
```

During initialization, you might use the Tab key to display these options. Then, you can choose which operating system to boot and enter it at the boot : prompt, as shown here:

```
LILO boot: <Tab>
linux  dos
boot:  dos
```

You have to be quick or LILO will default to using the first entry in this example after 50 seconds.

### **Fixing LILO**

Assume that you had a perfectly functioning Linux operating system until last night when you decided to add a Windows NT partition to the disk. Now, nothing will boot. What happened? Windows has overwritten the MBR.

How do you fix this problem? You must have a bootable DOS floppy at hand with `fdisk` available. Our recommendation is that you never use a DOS version older than version 5.0. Then, you use the DOS command `fdisk /mbr`. Many experienced administrators and users know when to perform this task, but often they do not know why.

The `fdisk /mbr` command will replace (not fix, but replace) the entire MBR with a clean DOS boot record. Yes, this situation means that the little LILO and NT program that previously lived in the MBR is now gone. The deletion is easily remedied because LILO can be recreated in two easy steps.

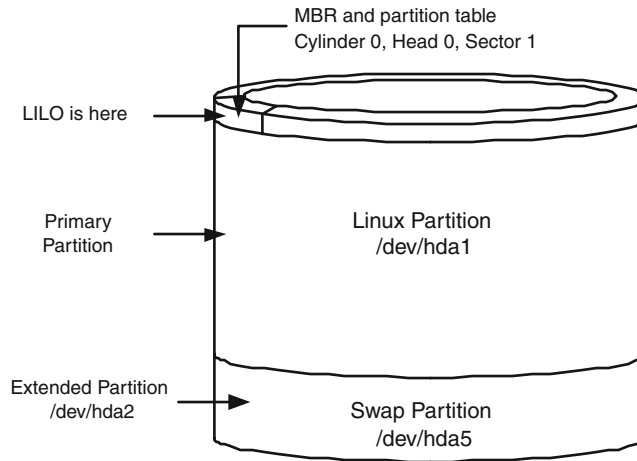
Boot your system with your rescue diskette. Make sure the `/etc/lilo.conf` file is exactly the way that you want it. Run the `/sbin/lilo` command to replace and copy the Linux code back into the MBR.

## Disk Partitioning: A Simple Approach

You can dedicate the entire hard drive space to Linux with one file system partition and one swap partition. There is nothing wrong with this approach. Red Hat distributions before version 7 did this technique. This action will enable any directory sharing the file system to grow freely. This situation is perfectly acceptable for either a desktop system or a small utility server that will not be used to store large amounts of user data. The reason we comment on that fact that this setup is not suitable for systems that are destined to store user data is that if the disk fills, your system's operating system is at risk. When the `/` root file system becomes 100 percent full, the system might come to an abrupt halt. It is our experience that when a system is shared with more than one person, no one watches the disk space. It is the same phenomenon where no one cleans the shared coffee room.

To introduce you to how this scenario works, we are going to start by looking at a simple disk partitioning scenario and move toward a more complex scenario. The simple disk configuration would look similar to the `fdisk` output shown next. Notice the use of logical device names to represent the various disk partitions. Device `/dev/hda1` with 19422533 byte blocks starting at 1 and ending at 2418 represents the `/` root file system. The root file system will have all of the directories in it. The `/dev/hda2` starting at 2419 and ending at 2482 represents the extended partition that contains the swap partition. We can see this partition because the swap partition also resides at 2419 through 2482. The swap partition is then represented by `/dev/hda5` and the extended partition is then represented by `/dev/hda2`. Another way of describing this scenario would be to say that





**Figure 2.5** A simple example of disk partitioning.

the swap partition is inside the extended partition. We could also say that the `/dev/hda5` partition is inside the `/dev/hda2` partition.

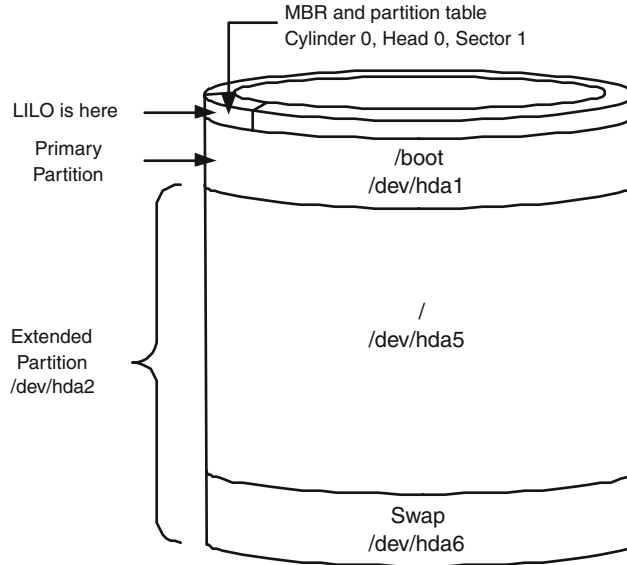
```
Disk /dev/hda: 255 heads, 63 sectors, 2482 cylinders
Units = cylinders of 16065 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	2418	19422553+	83	Linux
/dev/hda2		2419	2482	514080	5	Extended
/dev/hda5		2419	2482	514048+	82	Linux swap

Starting with Red Hat Distribution version 7.1, they changed the default disk partitioning behavior so that at least the boot files are now separate from the root file system files. This change is an improvement over previous configurations. This system will now protect the system boot files from disk full errors generated in the root file system. The new default auto partitioning used now is shown in Figure 2.6. This message is what you will get if you allow Linux to perform auto partitioning during the workstation type installation:

```
Disk /dev/hda: 255 heads, 63 sectors, 1245 cylinders
Units = cylinders of 16065 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	7	56196	83	Linux
/dev/hda2		8	1245	9944235	5	Extended



**Figure 2.6** New default disk partitioning for a workstation.

/dev/hda5	8	722	5743206	83	Linux
/dev/hda6	723	853	1052226	82	Linux swap

In Figure 2.6, there are four logical partitions called /dev/hda1, /dev/hda2, /dev/hda5, and /dev/hda6. These logical partition names (see Table 2.1) were assigned by the installation process and have special meaning to Linux. These logical partitions are representative of the physical disk configuration inside the system you are installing.

The logical name hdax depicts a primary IDE drive whereas a *Small Computer System Interface* (SCSI) drive would read sdax. Whether the disk is IDE or SCSI, a numbering convention is associated with each of the physical or logical partitions that are predictable. In our example, /dev/hda1 represents a small partition and will display at the Linux command line as a file system called “/boot.” As its name implies, it is a primary partition on the primary master IDE drive. The output from this `df -h` command was taken from the system depicted in Figure 2.6.

```
[root@HostB /root]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda5       5.4G 1016M  4.1G  20% /
/dev/hda1       53M   3.5M   46M   7% /boot
```

**Table 2.1** Logical Partition Names

NAME	PARTITION
/dev/hda1	First primary partition, primary master IDE drive
/dev/hda2	Second primary partition, primary master IDE drive
/dev/hda3	Third primary partition, primary master IDE drive
/dev/hda4	Fourth primary partition, primary master IDE drive
/dev/hda5	First logical partition, primary master IDE drive
/dev/hda6	Second logical partition, primary master IDE drive
/dev/hdb1	First primary partition, primary slave IDE drive
/dev/sda1	First primary partition, SCSI drive 1
/dev/sda2	Second primary partition, SCSI drive 1
/dev/sdb1	First primary partition, SCSI drive 2
/dev/sdc1	First primary partition, SCSI drive 3

This `df` output can be read as follows. The logical partition called `/dev/hda5` is mounted on `/`. Likewise, the `/dev/hda1` partition is mounted on `/boot`. Note the mount points. Although these are separate entities, they still look like a flat file system. This coding is what a listing of the root file system would look like. You cannot tell that there is a logical division in partitions unless you were to run a command such as `df` that lists the partitions along with their associated mount points. Regardless of what the partitions and mount points are, they all work together to display as one flat directory structure that looks like the following:

```
[root@HostB /root]# ls -l /
total 145
drwxr-xr-x  2 root  root      4096 Jun 17 08:33 bin
drwxr-xr-x  3 root  root      1024 Jun 18 05:07 boot
drwxr-xr-x 15 root  root     81920 Jun 18 05:07 dev
drwxr-xr-x 36 root  root      4096 Jun 18 05:07 etc
drwxr-xr-x  3 root  root      4096 Jun 17 08:33 home
drwxr-xr-x  7 root  root      4096 Jun 17 08:32 lib
drwxr-xr-x  2 root  root    16384 Jun 17 08:21 lost+found
drwxr-xr-x  2 root  root         0 Jun 18 05:07 misc
drwxr-xr-x  4 root  root      4096 Jun 17 08:23 mnt
drwxr-xr-x  2 root  root      4096 Aug 23 1999 opt
dr-xr-xr-x 44 root  root         0 Jun 18 05:06 proc
drwxr-xr-x  5 root  root      4096 Jun 17 19:30 root
```

```

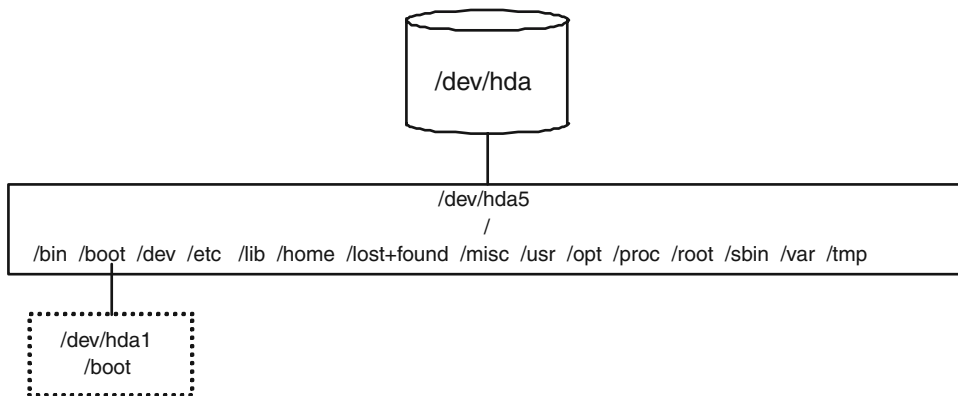
drwxr-xr-x    2 root    root        4096 Jun 17 08:33 sbin
drwxrwxrwt    8 root    root        4096 Jun 18 05:07 tmp
drwxr-xr-x   15 root    root        4096 Jun 17 08:24 usr
drwxr-xr-x   17 root    root        4096 Jun 17 08:37 var

```

An alternate way of looking at this particular scenario would be as follows. We have both `/dev/hda5` and `/dev/hda1` mounted. Once they are mounted, they are accessed as regular files and directories. In other words, if you copy a file from the `/boot` directory into `/tmp`, you are copying the file from `/dev/hda5` to `/dev/hda1`. Another very important point is that if your users fill the `/home` directories to 100 percent, that means that `/dev/hda1` is now 100 percent full as well and that all the other directories within `/dev/hda1` will be impacted. To ensure that this situation does not happen, we can assign `/home` to a separate partition just like `/boot`. We will show you how to do this task later in this section.

To reiterate, in this configuration, if the `/` file system fills to 100 percent, it will not affect the `/boot` file system. Yet another reason that separate partitions are used is to simplify tasks such as security and disk quota assignments. Many security features and utilities can be applied to a singular partition or mounted to a file system without affecting the others. Data mobility and back and restore procedures are also simplified.

Also, note that the `/dev/hda2` partition does not display with the `df` command because the `/dev/hda2` partition is an entity that is managed by the system. The `/dev/hda2` extended partition is the parent partition for both `/dev/hda5` and `/dev/hda6`. The swap partition known as logical partition `/dev/hda5` will not show up at the Linux command line either as a user-accessible file system. The reason is because the swap file system



**Figure 2.7** File system mounts for the workstation scenario.

should not be accessible to any user to write into and is accessed and maintained by the operating system itself.

The key to understanding how this scenario works is to look at a system that is already installed. The installation process itself creates a set of directories that are required. These directories and files are required for the operating system itself to function. You will notice that if you decide to install a server type system, you will end up with a series of partitions where each partition is subsequently mapped to a specific mount point. The following section shows the results of a server type installation.

## Disk Partitioning: A Complex Approach

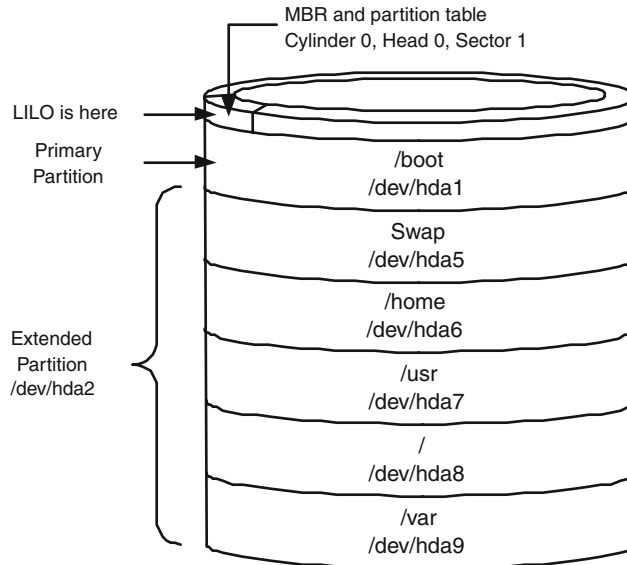
When installing servers, it is best practice to create separate file systems for the purposes of separating certain directories. Figure 2.6 is an example of a more complex disk partitioning scheme. The following scenario is an example of best practice using common conventions. Red Hat also adopted this configuration in version 7 to support its server type installation option. After choosing to install using the server type, your disk configuration would look like the following:

```
Disk /dev/hda: 255 heads, 63 sectors, 826 cylinders
Units = cylinders of 16065 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	7	56196	83	Linux
/dev/hda2		8	826	6578617+	5	Extended
/dev/hda5		8	203	1574338+	82	Linux swap
/dev/hda6		204	420	1743021	83	Linux
/dev/hda7		421	637	1743021	83	Linux
/dev/hda8		638	670	265041	83	Linux
/dev/hda9		671	703	265041	83	Linux

This configuration shows seven partitions. Five are user partitions and two are system partitions. This information is confirmed by observing the output for the `df -h` command, shown as follows:

```
[root@HostA /root]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda8        251M   52M  186M   22% /
/dev/hda1         53M   3.5M   46M    7% /boot
/dev/hda6        1.6G  476k  1.5G    1% /home
/dev/hda7        1.6G  804M  786M   51% /usr
/dev/hda9        251M   19M  219M    8% /var
```



**Figure 2.8** A complex example of disk partitioning.

The files and directories are contained within these partitions as shown. We have boxed the directories that correlate to the separate mount points in Figure 2.9 to help clarify the configuration. Figure 2.10 depicts the logical relationship between these physical entities.

You can see in Figure 2.9 that the directories that are shown in bold are all in the / partition, whereas the other directories (*boot*, *home*, *usr*, and *var*) all have their own partitions. The program will always behave this way. If we had chosen to create the partitions manually, we would have had more control over these directories. You could create as many of these that you deemed necessary. We will discuss the manual creation process shortly.

As a system administrator, you might not agree with the following—in which case you can create the necessary partitions that will complement your particular environment. For example, if we asked Disk Druid to create a partition for */tmp*, then it would have created a separate logical partition called */dev/hda10* with a mount point of */tmp*.

### Review of the Rationale for Separate Partitions

The following information is based on best practices. Whether or not to create a series of separate partitions will vary based on what it is you are trying to achieve. For those who are not familiar with the concepts the following guidelines are provided.

```
[root@HostA /root]# ls -l /
```

```
total 119
```

```
drwxr-xr-x  2 root  root   2048 Jun 17 15:16 bin
```

```
drwxr-xr-x  3 root  root   1024 Jun 19 09:11 boot
```

```
drwxr-xr-x 14 root  root  81920 Jun 19 09:13 dev
```

```
drwxr-xr-x 38 root  root   4096 Jun 19 09:11 etc
```

```
drwxr-xr-x  7 root  root   4096 Jun 17 23:17 home
```

```
drwxr-xr-x  7 root  root   3072 Jun 17 15:18 lib
```

```
drwxr-xr-x  2 root  root  12288 Jun 17 15:11 lost+found
```

```
drwxr-xr-x  2 root  root    0 Jun 19 09:11 misc
```

```
drwxr-xr-x  4 root  root   1024 Jun 17 15:12 mnt
```

```
drwxr-xr-x  2 root  root   1024 Aug 23 1999 opt
```

```
dr-xr-xr-x 52 root  root    0 Jun 19 02:11 proc
```

```
drwxr-xr-x  4 root  root   1024 Jun 18 14:38 root
```

```
drwxr-xr-x  2 root  root   3072 Jun 17 15:18 sbin
```

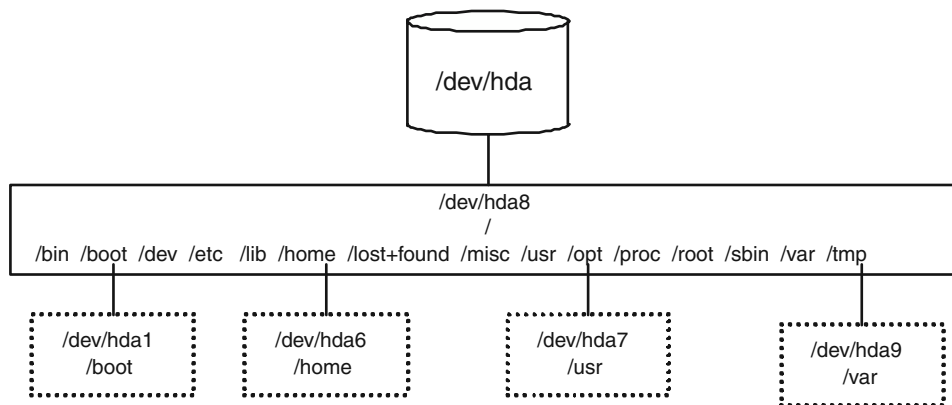
```
drwxrwxrwt 10 root  root   1024 Jun 19 09:18 tmp
```

```
drwxr-xr-x 16 root  root   4096 Jun 17 15:14 usr
```

```
drwxr-xr-x 20 root  root   1024 Jun 17 15:18 var
```

**Figure 2.9** File system files and directories after mounting.

To give */boot* its own logical file system partition. The */boot* contains the operating system kernel (vmlinux-2.4.2-2) along with a few other files used during the bootstrap process. Due to the limitations of most PC BIOSs, creating a small partition to hold these files is a good idea. You should make sure that the */boot* or */dev/hda1* partition stays inside the 1023 cylinder range unless you know for sure that your BIOS can handle locating a boot kernel outside this range (and



**Figure 2.10** File system mounts for the server scenario.

few can). This rule is called the “1024 cylinder rule” to facilitate system booting. Typically, the boot partition only needs to be between 20MB and 50MB maximum. It needs only to contain the files necessary to boot the system as well as some configuration files. It would be considered inappropriate to keep data files here.

**To give `/usr` or `/dev/hda7` its own logical file system partition.** This location is where most of the application software, commands, and libraries are installed. In the event that more applications are installed than you had originally expected, you will be able to easily back up and restore this file system partition. It is then possible to increase its size without having to touch any of the other file systems. Some system administrators will also further separate the application data by creating a `/usr/local` to keep their own scripts and utilities. Also, to create a `/usr/src`, it further isolates the Linux kernel sources and RPM software or their own software.

**To give `/home` or `/dev/hda6` its own logical file system partition to protect all other partitions from being accidentally filled to 100 percent capacity due to user error.** This partition is typically the most vulnerable because there are many users saving files to this area, and if no disk quotas have been enforced, then they can copy files and download excessive amounts of data to their home directories.

**To give `/var` or `/dev/hda9` its own logical file system partition.** Typically, this location is where most applications and the operating system logs accumulate. These logs can fill a file system very quickly due to irregular system activity if not monitored regularly.

**To create multiple swap partitions for performance and/or maintenance reasons.** Large systems may need an entire drive dedicated to swapping. Whereas in a smaller system, the swap partition should be placed on a drive where it will not have to compete with active file or database reads and writes. In either case this is an issue which should be addressed during the planning stage.

All other directories are typically placed into `/`, where they will all share the same partition. We are now going to take this configuration a step farther and involve the use of addition disks as well as separate partitions. It is not unusual for a system administrator to choose to isolate a partition on a specific disk for management purposes. This type of configuration is very common for large database servers, Web servers, and application servers. For



example, if the data and application files are all contained on a separate disk, then that disk can be easily removed and placed in a new system and brought online quite easily. If the disk is hot swappable, it only takes a few minutes.

### ***Multiple Drive Configuration Considerations***

Drive terminology changes from operating system to operating system. Linux, too, has its own protocol for recognizing and labeling drives. Knowing what the default behavior should be will help when building and configuring systems. An installation expert will plan ahead and know based on experience what the drives eventually will be called and how their respective partitions will be labeled after performing the installation. Then when the installation is complete, if the drives are not labeled correctly then perhaps a cabling issue needs to be rectified.

#### **IDE Drive Geometry**

The naming conventions for the IDE drives and devices shown in Table 2.2 indicate respective behaviors that in turn are reflections of how they will be labeled by Linux in the target system. Newer IDE device configuration rules enable up to four channels and not just two in a single system, each with two devices attached. The devices do not have to be configured consecutively, but this practice is helpful in understanding how the system works. Problems might occur with older or existing systems in which

**Table 2.2** IDE Drive Geometry

CHANNEL	JUMPER	HDX
ide0	master	hda
ide0	slave	hdb
ide1	master	hdc
ide1	slave	hdd
ide2	master	hde
ide2	slave	hdf
ide3	master	hdg
ide3	slave	hdh

Note: ide0 = primary; ide1 = secondary; ide2 = tertiary; and ide3 = quarternary.

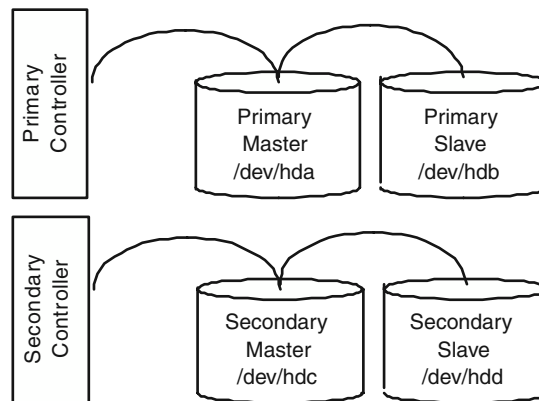
older IDE drives and devices are attached to newer IDE controllers and vice versa. Remember to test and document what you have done to avoid installation headaches.

The partition numbering scheme follows an old PC standard in which each hard drive was limited to four primary partitions. Any one of the four partitions can also be designated as an extended partition. Within the extended partition, up to 12 logical partitions can be further specified.

Most motherboards typically come with four IDE ports that can be configured as follows (see Figure 2.11).

This feature will allow you many different configurations. We will only describe a few. One very popular workstation configuration is to have one hard disk and one CD-ROM. In this case, it would be better to cable your hard disk to the primary controller and cable your CD-ROM to the secondary controller both as primary devices: `/dev/hda` and `/dev/hdc`, respectively. This situation would be preferable to having both devices cabled to the primary as master and slave: `/dev/hda` and `/dev/hdb`. This recommendation is based solely on utilization of the controllers.

**TIP** To prove or observe your configuration after installation, go to the `/proc/ide` directory and you should see a directory for each device that you have installed. For example, there should be a directory called `/dev/hda` and `/dev/hdc`. Inside the `/proc/ide/hda` there should be some files that you can list to confirm the details that are specific to the devices that you have. If you find that these settings are incorrect based on your drive documentation, then you have either BIOS or master/slave configuration issues.



**Figure 2.11** Typical IDE disk configuration.

If you are planning to mirror your drives, you would be wise to mirror /dev/hda with /dev/hdc and not /dev/hda with /dev/hdb. First, performing parallel writes through two controllers is safer and faster. Second, if the one controller that both your drives are attached to fails, then everything is unavailable.

The only other situation that requires thought is how to handle two hard drives and a CD-ROM. Is it better to put both the hard drives on the primary controller as /dev/hda and hdb? Alternatively, would it be better to have one hard drive and the CD-ROM off the primary controller and put the second hard disk off the secondary controller? Well, it is up to you. If you were mirroring, we would suggest the former. If your CD-ROM were used constantly, then you would probably want to use the later suggestion. Whatever you do, make sure that the devices are configured correctly as master or slave.

All of this discussion assumes that all the drives and CD-ROMs involved are all compatible technology. There are different kinds of IDE technology, and as discussed in Chapter 1, "Preparing for Installation: Chronicling Your Equipment," some should not be mixed. For example, do not penalize a hard drive by having it share a controller with a very slow CD-ROM device.

### SCSI Drive Geometry

The naming conventions for SCSI drives and devices shown in Table 2.3, like the IDE drives, indicate respective positions within the system. The table shows that all the devices are all participants on channel 00, and they are all of the same SCSI specification.

A listing of the `/proc/scsi/scsi` file confirms that these drives are all configured on a single channel called `scsi0`. Each device is configured at its own `scsi` Id, respectively.

```
[root@HostC scsi]# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
```

**Table 2.3** SCSI Drive Geometry

CHANNEL	DRIVE	SCSI ID
00	sda	00
00	sdb	01
00	sdc	03
00	sdd	04

```

Vendor: COMPAQ   Model: DCHS09Y           Rev: 5858
Type:   Direct-Access                     ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 01 Lun: 00
Vendor: COMPAQ   Model: DCHS09Y           Rev: 5858
Type:   Direct-Access                     ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 03 Lun: 00
Vendor: COMPAQ   Model: DCHS09Y           Rev: 5858
Type:   Direct-Access                     ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 04 Lun: 00
Vendor: COMPAQ   Model: DCHS09Y           Rev: 5858
Type:   Direct-Access                     ANSI SCSI revision: 02

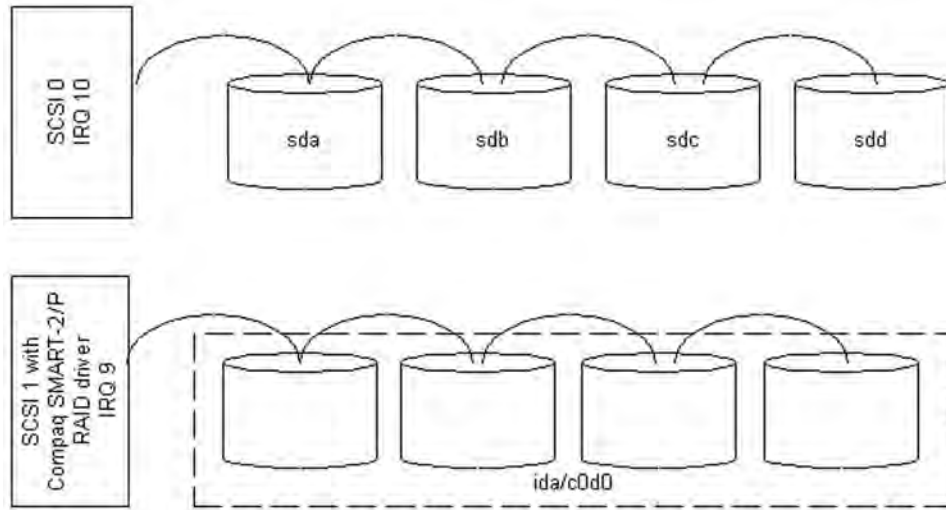
```

There has been a succession of the SCSI standards developed since 1986, which resulted in many different types of SCSI devices being available. Although the standards are more mature and are better understood now, occasionally they are mistakenly combined. We have noticed a trend to use older SCSI equipment to assemble Linux utility servers. The reason might be that what used to be a main server is now being rotated and salvaged into a Linux utility server. As a result, we have had to solve many issues regarding the use of SCSI devices. Most of the issues have evolved around an unhealthy combination of SCSI I, II, and III technology being installed on the same channel in the same system. These differing SCSI standards do not all interoperate successfully. In all these cases, removing the older SCSI device from the SCSI channel has made it work. The main reason is that as the standards were developed, the speed and protocol capabilities of the devices were improved (changed)—thus, older SCSI equipment is incompatible with newer SCSI equipment (with a few exceptions). Keep this fact in mind if you are attempting to build a Linux server out of older, used SCSI equipment that has been rotated from other systems.

### Approaches to Configuring RAID

There are two basic approaches: software and hardware. Linux does have a software *Redundant Array of Inexpensive Disks* (RAID) solution, but it is awkward and very resource intensive and has nothing in common with hardware RAID. The software solution has a very limited list of supported controllers and uses much of the system's memory and CPU to manage the writes. For this reason, hardware RAID solutions have always been preferable.

The hardware RAID solution is vendor specific. For example, we have demonstrated a Compaq solution here. Our solution was to build on a Compaq Proliant with a Compaq RAID controller and Compaq drives. To set up this system, we had to visit [www.compaq.com](http://www.compaq.com) and download all of the necessary Red Hat Linux support software specific to the distribution



**Figure 2.12** The SCSI disk configuration we used.

before installing Linux to set up the Compaq system. There are many ways to configure SCSI devices. Our configuration is documented in Figure 2.12.

This situation will hold true for any manufacturer that supports Linux RAID. There is nothing on the Linux distribution CD-ROMs to set up a system for hardware RAID. Instructions for configuring the RAID controller and RAID systems in general are outside the scope of this document and are vendor specific, so we suggest that you visit that vendor's Web site for these specific instructions. We mentioned this fact because we did not want you to think that our SCSI RAID installation was automatic and used tricky code and magic. More information on how to install Linux on these drives is included in the next chapter.



## Installing Linux from a CD-ROM

In this chapter, we discuss the necessity for a boot disk and then walk through Linux installations on two different system architectures: IDE and SCSI. First, you will need a copy of Linux. You can obtain Linux from several sources, including your local software vendor, the Internet, or on the Red Hat Distribution CD-ROM included with this book. If you have lots of bandwidth and a CD-writer, you can actually download the ISO images directly from Red Hat. If you are interested in this method, go to [www.redhat.com/download/howto\\_download.html](http://www.redhat.com/download/howto_download.html) for further instructions.

The installation menus are dynamic and might change order for various distribution versions. The installation in the chapter uses Red Hat Distribution 7.1, which implements a Linux kernel level of 2.4.2-2. All the Red Hat Distributions starting at version 6.X offer a GUI installation option. We will bypass the GUI installation and intentionally guide you through a text menu installation for three reasons. First, there is a high failure rate for the GUI interface. Second, our objective is to present you with a more consistent view of a Linux installation procedure that has been in place since the very early versions. Finally, the text menu installation includes some

menus that you would not normally see in the GUI install, thus giving you a more technical view of the overall procedure.

By definition, “Linux is a UNIX-like operating system kernel that can be freely distributed.” Technically, Linux is not freeware and is *not* in the public domain. Linux is protected by both a code of honor and GPL (GNU Public License), which essentially means that you should not (and cannot) change the source code of the kernel and release it under a more restrictive license. Any changes or modifications made to the Linux kernel are governed by the GPL, as well. In other words, you can get the code from anywhere and legally use it, but do not change it and resell it for profit.

## Initiating a Linux Installation

---

Technically, all Linux CDs are bootable. Whether your Linux CD will boot your system depends upon whether the CMOS on your PC is set correctly. For example, most PCs are configured to boot from floppy first and from hard disk second. You have to change the boot sequence to CD-ROM first, floppy second, and hard disk third. If it stills fails to boot, then your BIOS or hard disk controller probably does not support booting from a CD-ROM—in which case you will have to create a boot diskette. You can create a boot disk from either a DOS/Windows or UNIX session. You should have two diskettes available to accomplish this task.

In some Linux versions before 6.x, to provide additional support for *Personal Computer Memory Card Internal Association* (PCMCIA) and *Session Message Block* (SMB) installation, you have to create additional diskettes ahead of time. You will be prompted for these diskettes, and they have to be ready. With newer versions of Red Hat Linux, however, the installation procedure has been changed—so this issue no longer exists.

Meanwhile, you will be prompted to insert a blank diskette, which will be used to create a customized rescue diskette. We highly recommend that you do so. For now, though, let’s return to our discussion of boot disks.

The method for creating a Linux install boot disk depends on the operating system you have to begin with and whether you want to use that operating system to create the boot disk. We discuss two methods: first, creation from within a DOS/Windows operating system; and second, from within a UNIX system. Boot diskettes made using DOS/Windows or

Linux will behave the same way; thus, choosing one over the other makes no difference.

## Creating the Install Boot Disk from a DOS/Windows System

To create a boot disk from DOS/Windows, we will use a program called `rawrite`. In this example session, we insert the Red Hat Linux product CD-ROM into the CD-ROM drive of a DOS or Windows system. The `rawrite` utility will be in a directory called `dosutils`. The boot file that you need is `boot.img`. Your listing of the `images` directory might not be identical but should resemble the following. When you run `rawrite`, it will prompt you for the `boot.img` file to transfer to the boot diskette. Use the following instructions to create a boot diskette from a DOS or Windows system:

### Step 1

Put the Red Hat CD in the CD-ROM drive. Change to the `/images` directory and list the contents:

```
G:
G:>cd \images
G:\images>dir
Volume in drive G is Red Hat Linux_i3
Volume Serial Number is 3BC3-AD45

Directory of G:\images

08/30/00  06:40p      <DIR>      .
08/30/00  06:44p      <DIR>      ..
08/30/00  06:48p                485 TRANS.TBL
08/30/00  06:39p            1,474,560 boot.img
08/30/00  06:39p            1,474,560 bootnet.img
08/30/00  06:40p      <DIR>      de
08/30/00  06:40p            860,160 drivers.img
08/30/00  06:40p      <DIR>      es
08/30/00  06:40p      <DIR>      fr
08/30/00  06:40p      <DIR>      it
08/30/00  06:40p            147,456 paride.img
08/30/00  06:39p            1,474,560 pcmcia.img
08/30/00  06:40p      <DIR>      pxeboot
08/30/00  06:40p      <DIR>      sv
               14 File(s)            5,431,781 bytes
                                0 bytes free
```



## Step 2

Run the `rawrite` utility from the `\dosutils` directory and transfer the *boot.img* file to the floppy disk:

```
G:\dosutils\rawrite
Enter disk image source file name:  boot.img
Enter target diskette drive:  a:
Please insert a formatted diskette into drive A: and press -ENTER- : <CR>
```

The result will be a bootable floppy with the following files on it that you can use to boot the system and to initiate the installation process:

```
Volume in drive A is LINUX BOOT
Volume Serial Number is 2410-07EF

Directory of A:\

08/30/00  06:39p                5,860 LD LINUX.SYS
08/30/00  06:39p                425 SYSLINUX.CFG
08/30/00  06:39p            4,807 TEMPLATE.IMG
08/30/00  06:39p          789,543 INITRD.IMG
08/30/00  06:39p          600,265 VMLINUX
08/30/00  06:39p             751 BOOT.MSG
08/30/00  06:39p             653 EXPERT.MSG
08/30/00  06:39p             859 GENERAL.MSG
08/30/00  06:39p             860 PARAM.MSG
08/30/00  06:39p             506 RESCUE.MSG
08/30/00  06:39p             545 SNAKE.MSG
          11 File(s)          1,405,074 bytes
                               49,664 bytes free
```

## Creating the Install Boot Disk from a UNIX System

In a UNIX-based environment, you can use the `dd` utility to create a boot diskette. First, you have to mount the CD-ROM to get to the *images* directory, where the *boot.img* file resides. The `dd` utility is a UNIX utility, so it is not on the CD.

Put the CD in the CD-ROM drive, mount the CD-ROM, and list the contents of the *images* directory. The boot file that you need is *boot.img*. Your listing of the *images* directory might not be identical but should resemble the following. When you run `dd`, you will supply the *boot.img* file as an argument to transfer to the boot diskette. Use the following instructions to create a boot diskette from a UNIX system:

**Step 1**

Put the Red Hat CD in the CD-ROM drive. Change to the */images* directory and list the contents:

```
Red Hat Linux release 7.0 (Guinness)
Kernel 2.4.2-2 on an i686
login: linda
Password:
Last login: Thu Jun  7 11:48:36 from 8.3.105.200
[linda@HostA linda]$ su - root
Password:
[root@HostA /root]# mount /dev/cdrom
[root@HostA /root]# mount
/dev/hda1 on / type ext2 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
automount(pid469) on /misc type autofs
(rw,fd=5,pgrp=469,minproto=2,maxproto=3)
/dev/hdc on /mnt/cdrom type iso9660 (ro,nosuid,nodev)
[root@HostA /root]# cd /mnt/cdrom
[root@HostA cdrom]# ls -l
total 63
-rw-r--r--      10 root      root          18385 Sep  7  1999 COPYING
-rw-r--r--      10 root      root           4730 Aug 25  2000 README
-rw-r--r--       5 root      root        25206 Aug 27  2000 RELEASE-NOTES
-rw-r--r--      10 root      root          1908 Sep 25  1999 RPM-GPG-KEY
drwxr-xr-x       4 root      root          2048 Aug 30  2000 RedHat
-r--r--r--       1 root      root           465 Aug 30  2000 TRANS.TBL
-r--r--r--       1 root      root          2048 Aug 30  2000 boot.cat
drwxr-xr-x       6 root      root          4096 Aug 30  2000 dosutils
drwxr-xr-x       8 root      root          2048 Aug 30  2000 images
```

**Step 2**

Change to the */mnt/cdrom/images* directory and list the contents of that directory to confirm that the *boot.img* file is present:

```
[root@HostA cdrom]# cd /mnt/cdrom/images
[root@HostA images]# ls -l
total 5333
-r--r--r--       1 root      root           485 Aug 30  2000 TRANS.TBL
-rw-r--r--       2 root      root       1474560 Aug 30  2000 boot.img
-rw-r--r--       2 root      root       1474560 Aug 30  2000 bootnet.img
drwxr-xr-x       2 root      root          2048 Aug 30  2000 de
-rw-r--r--       2 root      root      860160 Aug 30  2000 drivers.img
drwxr-xr-x       2 root      root          2048 Aug 30  2000 es
drwxr-xr-x       2 root      root          2048 Aug 30  2000 fr
```

```
drwxr-xr-x      2 root    root          2048 Aug 30  2000 it
-rw-r--r--      2 root    root        147456 Aug 30  2000 paride.img
-rw-r--r--      2 root    root       1474560 Aug 30  2000 pcmcia.img
drwxr-xr-x      2 root    root          2048 Aug 30  2000 pxeboot
drwxr-xr-x      2 root    root          2048 Aug 30  2000 sv
```

### Step 3

Run the `dd` utility to transfer the *boot.img* file to the floppy diskette:

```
[root@HostA images]# dd if=boot.img of=/dev/fd0
2880+0 records in
2880+0 records out
```

The result will be a bootable floppy with the following files on it that you can use to boot the system and to initiate the installation process. Please note here out of interest that although we used the `dd` utility, the files are identical to those created with `rawrite`:

```
[root@HostA /root]# mount /dev/fd0
[root@HostA /root]# cd /mnt/floppy
[root@HostA floppy]# ls -l
total 1401
-rwxr-xr-x      1 root    root          955 Apr  8 22:37 boot.msg
-rwxr-xr-x      1 root    root          658 Apr  8 22:37 expert.msg
-rwxr-xr-x      1 root    root        1202 Apr  8 22:37 general.msg
-rwxr-xr-x      1 root    root       768551 Apr  8 22:37 initrd.img
-r-xr-xr-x      1 root    root        6192 Apr  8 22:37 ldlinux.sys
-rwxr-xr-x      1 root    root          862 Apr  8 22:37 param.msg
-rwxr-xr-x      1 root    root          506 Apr  8 22:37 rescue.msg
-rwxr-xr-x      1 root    root          608 Apr  8 22:37 syslinux.cfg
-rwxr-xr-x      1 root    root       652144 Apr  8 22:37 vmlinuz
```

**TIP** Red Hat has reported certain problems with boot images. Consider downloading the latest recommended boot images from [www.redhat.com/errata](http://www.redhat.com/errata). If you are determined to perform a GUI-assisted installation and your video is not working, consider downloading a newer *boot.img* file. That might resolve the problem.

## Invoking the Linux Installation Program

The Linux installation program consists of a series of dialog boxes sometimes called *screens* or *windows*. The dialog boxes present information to

you and request information and decisions from you. The two most common methods used to invoke the installation program are as follows:

- Inserting the Linux CD-ROM into the CD-ROM drive and booting the system (providing that the system CMOS has been set to facilitate booting from the CD-ROM drive)
- Inserting a bootable floppy disk into the floppy disk drive and booting the system (providing that the system CMOS has been set to facilitate booting from the floppy disk drive)

The install chronicled in this section presumes that you will use the floppy boot method. In fact, you will eventually see a dialog box that instructs you to insert the CD.

## Entering Information in the Installation Program

As you install Linux, you will notice several different ways to enter information into the dialog boxes. These methods are as follows:

- Test input.** You will encounter dialog boxes wherein you are to type information, usually on a dashed line. This information insertion method is used in conjunction with other methods such as buttons, toggles, and so on.
- Check boxes.** These are spaces defined by brackets or boxes, which you will either select or deselect by pressing the space bar.
- Buttons.** These are generally square boxes that you select with the Tab key and then select by pressing the Enter key.
- Scroll and select.** Some dialog boxes contain lists of components, packages, or default services from which you will choose by scrolling up or down and selecting. You will use combinations of the Tab and space bar keys, followed by pressing Enter at the end of the dialog boxes.

We present this list here for two reasons. First, there are the insertion methods you have to use, and this listing serves as a guide. Second and more importantly, the list helps you focus on the issue at hand rather than on the collection of information. If you occasionally make a mistake, take heart—you will not be the first or the last to do so.

Insert your CD-ROM and boot the system. If you have trouble booting, read the following paragraphs. If you do not experience trouble booting the system, proceed to Step 1.

## Installing Linux on an IDE System

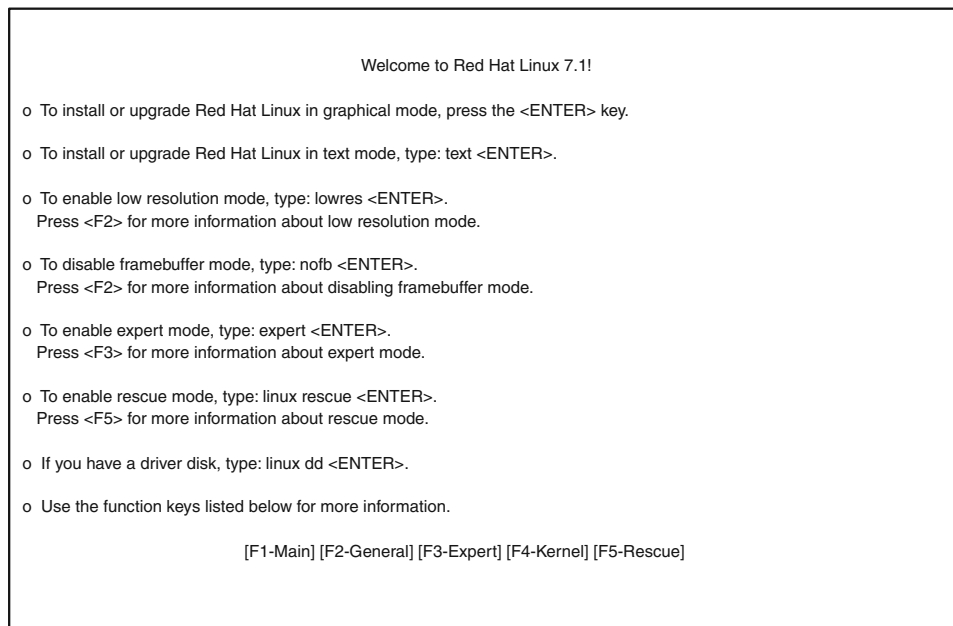
---

In this section, Linux is installed on a system with an *Integrated Drive Electronics* (IDE) architecture. IDE is an IBM PC *Industry Standard Architecture* (ISA). It is generally less expensive than SCSI and is simpler to configure. We have drawn the hard disk configuration for you to study the partitions and mount point data (see Figure 3.1).

A network connection to the Internet is required; therefore, we have obtained some IP addresses.

### ***CD Booting Problem Determination***

Sometimes, it appears that Linux cannot see or boot from IDE CD-ROMs. If this situation occurs, you should try to determine why Linux could not see the IDE device. It is possible that the BIOS cannot see or read an IDE



**Figure 3.1** IDE disk configuration.

channel because the BIOS is limited. This situation is probably the case if the system was functioning properly before under a different operating system. The solution is to either upgrade the BIOS or create a floppy boot diskette. We would recommend that you create a floppy boot diskette because it is easier. Another reason might be that the system might not be configured correctly. This situation is usually the case if you have taken the opportunity to add or move drives around in the system. Check the master/slave jumpers on all the drives and devices. If the drives are all chained off of the same IDE controller, then one device should be configured as the master and the other device should be the slave. You will have to locate the documentation for the IDE device to find out how the jumpers are to be arranged, because these instructions vary for different devices.

Another more interesting problem is dealing with a CD-ROM that will not load automatically after the installation. To get the system up and running, you can enter the following statement at the LILO prompt during the boot sequence:

```
boot: linux hdx=cdrom
```

*x* is the IDE letter that Linux specifies for the drive. This letter will vary depending on the IDE bus to which the drive is attached. This scenario might also cause you some problems during boot time. Make a note of it, and if the system also fails to boot after the installation, then revisit your cabling and master/slave configuration.

### ***Troubleshooting the IDE Installation***

If you are having trouble with the installation, stop and research what you are doing. Based on our experience, most problems involve hard drive configurations or hardware compatibility, followed by network card and bad driver configuration.

With respect to drives, the problem is often a lack of documentation on either the jumper or the termination settings. If you have a pair of IDE drives or a hard drive and CD-ROM combination, it is usually a master/slave configuration issue. If that is correct, then make sure that all the IDE devices are compatible with each other. In other words, do not mix older IDE technology with the newer, enhanced IDE technology. If they work together, then you are lucky. We have a saying in the industry, "Are we being lucky or smart?"

Network card and driver combinations can also be troublesome. It is our opinion that there has been amazing progress in this area over the past two

years. Three years ago, there were very few network cards supported in Linux—and now there are many. The newer network cards are PCI and therefore are self-configuring. When it works, it is great. Of course, how successfully this equipment works with your system is entirely dependent on the BIOS that you have installed on your system. If the BIOS are buggy or old, then this situation could be very difficult. Do not ignore the DOS utilities that we told you about in Chapter 1. Do not spend very much time on a network card that does not work without checking out the IRQ, I/O port, and memory address space that it is using—PCI or no PCI.

Watch for the following nasty behavior. You can ping the card internally, but it will not respond to a DHCP server. This situation is typical of a resource conflict, usually memory space. To check these issues, go directly to the */proc* directory and list the contents of all files called *iomem*, *ioports*, *irq*, and *net*. If the network card is not even present in these files, then the *irq* on the network card is no doubt set to something that is already used by another device. If you cannot even ping the card internally, then the driver is probably not functioning correctly. If you do not get a link light on the back of the network card, then it is probably the wrong driver or network card itself. We recommend that you only spend so much time with a network card that will not work and then go out and buy one. The last D-Link network card that I purchased was less than \$20. Your time has to be worth more than that. Barring that approach, you can find many Linux drivers at [www.driversguide.com](http://www.driversguide.com).

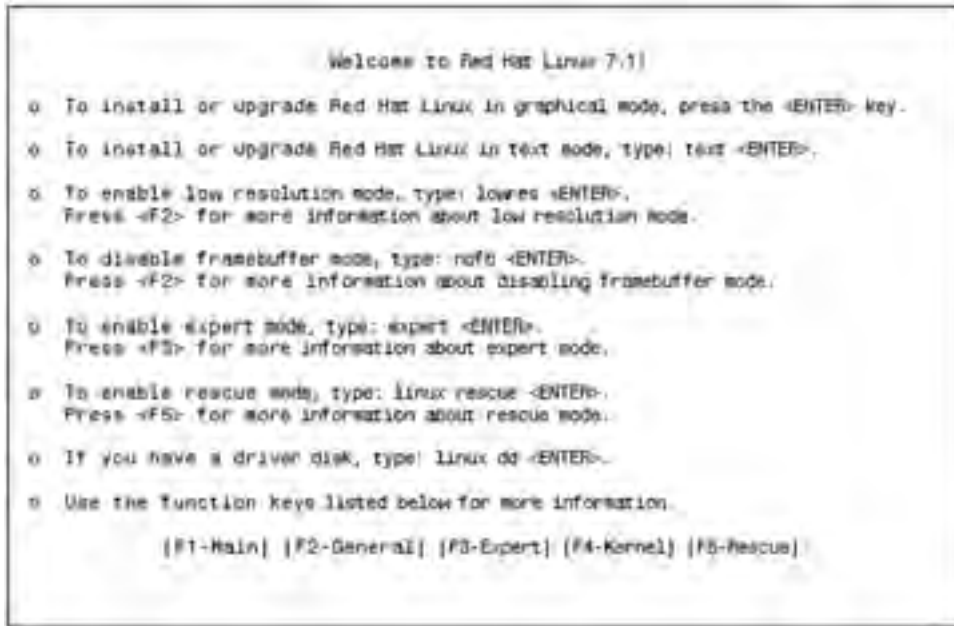
Video configuration is also a huge concern—so huge that we dedicated an entire chapter to it. Please refer to that chapter for advice on video cards.

## Stepping through the Installation

Our screens do not necessarily reflect what you will see on your system. They will, however, give you an excellent idea of what information you will require for your installation. We will explain default behavior and our rationale for choosing specific options. It is our intention that you will be able to make informed decisions when performing your own installation.

### Step 1. Boot Message

If you were to walk away from the first screen without selecting anything, the installation program would proceed to boot the system on its own and would default to the GUI install mode. We are specifically going to demonstrate using *text* mode.



**Figure 3.2** Welcome to Red Hat Linux 7.1! (boot message).

**Install or upgrade . . . graphical mode.** This option is the default, and it will launch a GUI install panel. Be aware that this choice has a high video failure rate because it does not interpret all video cards and monitors correctly. If you choose this option and it fails, we recommend that you immediately reboot and use the text option.

**Install or upgrade . . . text mode.** For practical reasons and in order to facilitate the installation of the majority of systems with no graphical support, use this option. Just a reminder, though—not all systems destined to be servers will have graphical support.

**Enable expert mode.** This option is not for the faint at heart but is sometimes necessary—for example, when Autoprobe does not appear to be discovering your card settings correctly. These dialog boxes will enable you to supply the necessary information for all card settings, such as IRQ, I/O, and DMA.

**Enable rescue mode.** This feature will enable you to boot a system from CD-ROM. This option is a last resort if it will not boot the system and you do not have a rescue diskette. You can create a rescue disk after the installation is finished should you have skipped this



step during the installation. To create a post installation rescue diskette, go to the `/lib/modules` directory and use the following command to create a rescue diskette customized for your system:

```
#cd /lib/modules
#mkbootdisk --device /dev/fd0 kernel_number
```

`kernel_number` is in the format 2.4.2-2. When you choose which kernel you want to install, you will insert the appropriate version numbers for the level of kernel you are currently running. You should always have a rescue diskette on hand that is specific to each system. For example, it is considered bad practice to use a rescue diskette from a system with a kernel level of 2.2.14-5 to rescue a system that is supposed to be at a kernel level of 2.4.2-2. The current kernel level of any system can be determined by observing the login prompt or by using the `uname -a` command and the system prompt.

You can also create a rescue diskette directory from the Red Hat CD by going to the `/images` directory and issuing the following command:

```
G:\>
G:\>cd images
G:\images>dd if=rescue.img of=/dev/fd0 bs=1440 count=1
```

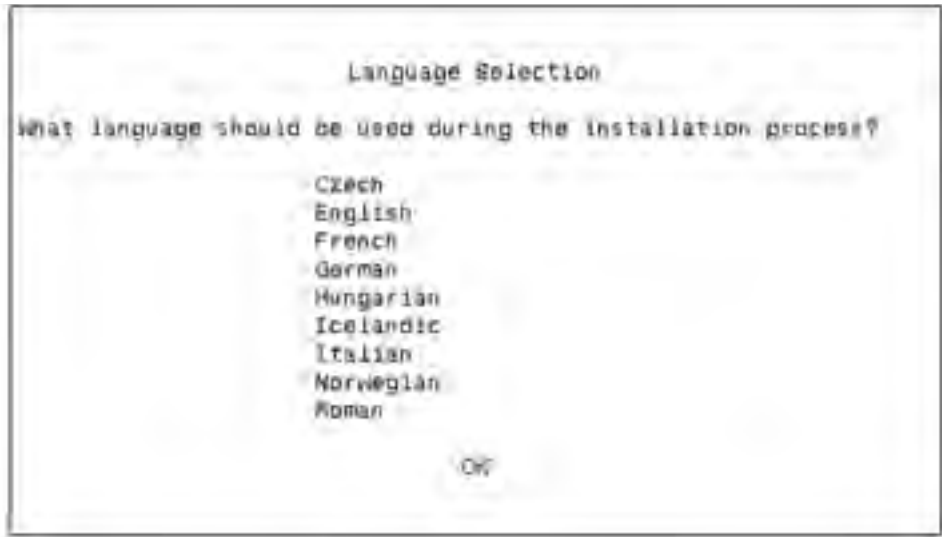
All the rest of the dialog boxes presume that you have chosen text mode. We chose text mode.

### Step 2. Select Language

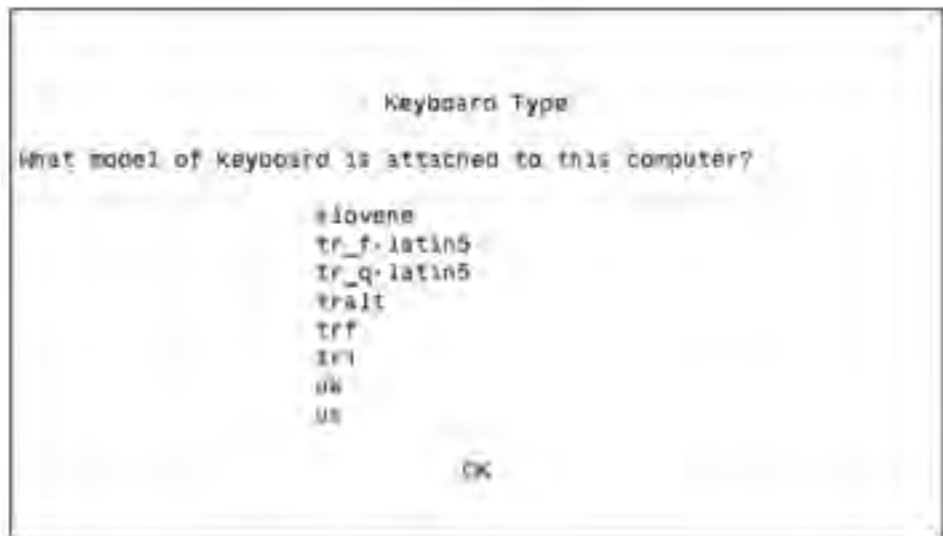
Whatever language you choose here will become the default language used for the system. You can add another language later should you require a multilingual system. In Figure 3.3 we chose English because this system is going to be based in North America and we have no additional language requirements.

### Step 3. Specify Keyboard

Do not be creative here. This question is a technical question about keyboard drivers. If you answer this question incorrectly, then issues might arise when you try to enable and map keys. Not all keyboards are created equal. You can, however, choose the closest one here and adjust it later when the system is up and running. In Figure 3.4 we chose us (United

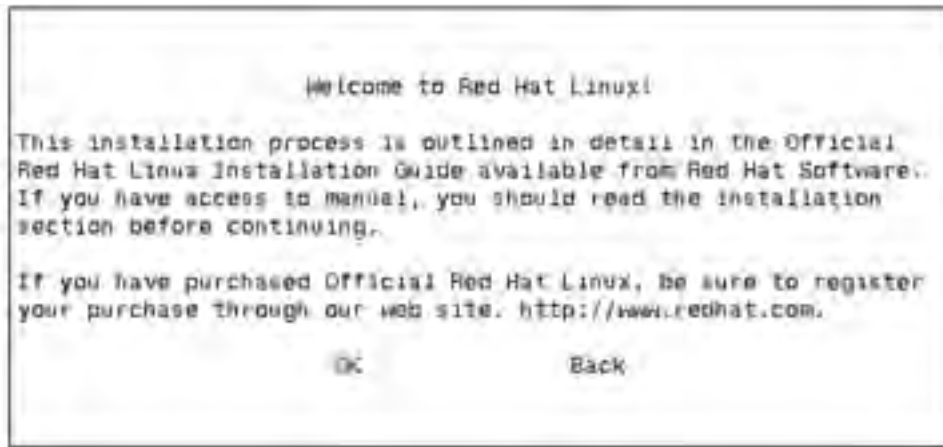


**Figure 3.3** Language selection.



**Figure 3.4** Keyboard type.

States) even though this system will primarily function within Canada. This is because there are fewer compatibility issues with US English command and character set support for the application software that we use.



**Figure 3.5** Welcome to Red Hat Linux! (installation).

#### **Step 4. Welcome**

After reading the text presented to you in Figure 3.5, press OK and go to the next screen.

#### **Step 5. Specify Installation Type**

Two the most popular features of contemporary Linux systems are the new desktops, GNOME and KDE. The one you choose should depend on what you want to do. If you are experimenting, try to install one and then the other. GNOME can run KDE applications and vice versa. Consider your choice between the two as making one dominant over the other. Of course, because the industry is sometimes political, the KDE development group has different mandates than the GNOME development group. Thus, we have two different factions creating desktops and utilities for such desktops.

Notice that both the GNOME and KDE choices here are workstation solutions. In other words, the default suggested combination of packages and utilities is well suited for a desktop system such as that for an end user. An end user here would also include a code developer or system administrator, whereas the server solution offered here would install a set of packages and utilities that you would not necessarily want on a workstation. For your information, if you choose Server System, no X server will be installed. This situation is a change from previous version behavior. If you wish to have an X server for a server class system, visit the Select individual packages during the installation or use the rpm utility after the installation is complete.

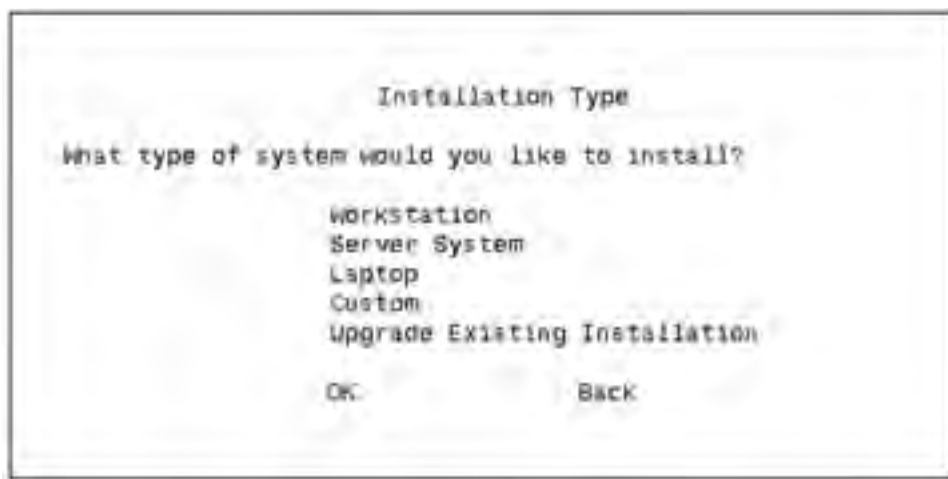
We are going to demonstrate a *custom* installation, which will enable you to specify exactly which packages and utilities you need. Just as a point of interest, should you decide to install absolutely all of the packages, you will have to create a 2GB system partition to hold the files (which does not include room for any data).

Finally, they have added a choice for laptop computers. This option is in response to so many users trying to get Linux up and running on their laptops, only to find out that they were missing many of the required packages. The default installation for a laptop now includes those packages such as a PCMCIA slot and power and TFT video support utilities.

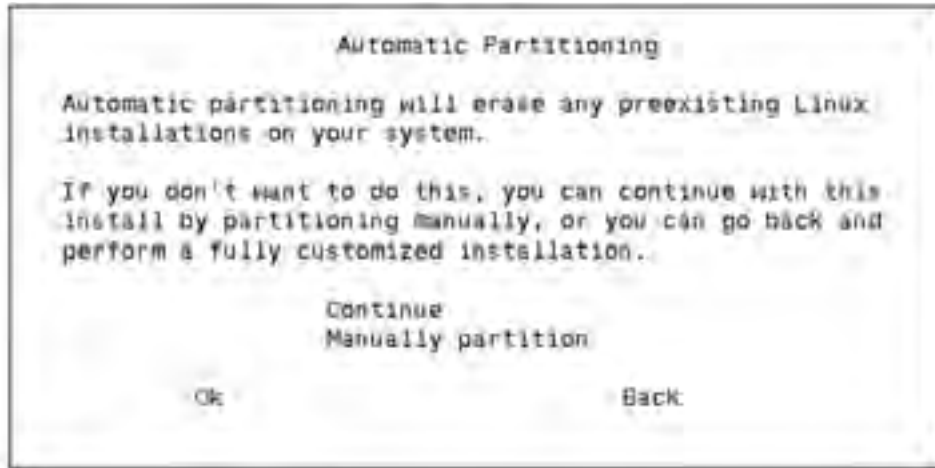
There is also an upgrade path available. We highly recommend that you use this feature to upgrade a Linux distribution or Linux kernel to the next level. This action is far preferable to attempting to recompile a kernel. Recompiling kernels is a specialized task that we are going to address in another chapter in this book. In Figure 3.6 we chose Custom so that we could customize the list of packages to be installed for base functionality.

### Step 6. Partitioning

In Figure 3.7 we are going to Continue to use automatic partitioning here to show how the system will look after the installation. The SCSI installation will use a complex approach using Disk Druid, whereby we will split the file system directories into separate, discrete partitions. The automatic partitioning will create a small partition called `/boot`, which will contain



**Figure 3.6** Installation type.



**Figure 3.7** Automatic partitioning.

the boot files for the system. All other files and directories will go into one large partition called `/`.

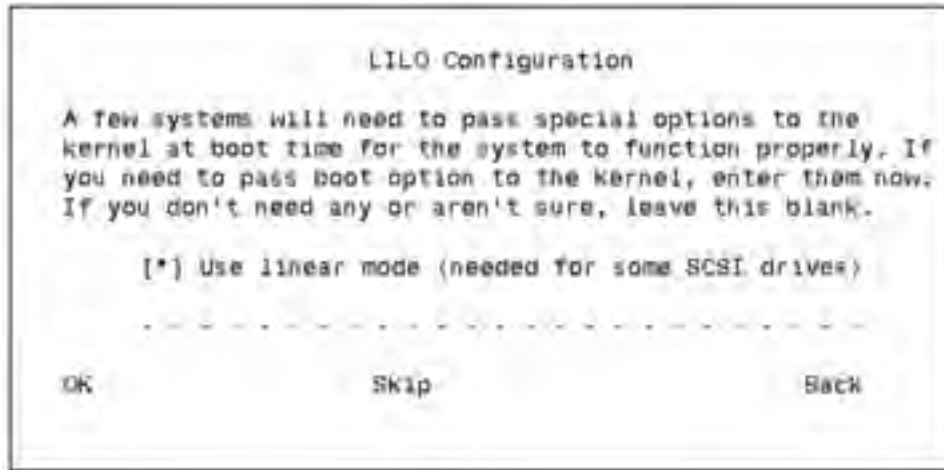
Note also that this installation is using Red Hat 7.1, which handles this function differently from the previous Red Hat 7.0 distribution (where all the files including the boot files would go into one large partition called `/`).

### **Step 7. Installing and Configuring LILO**

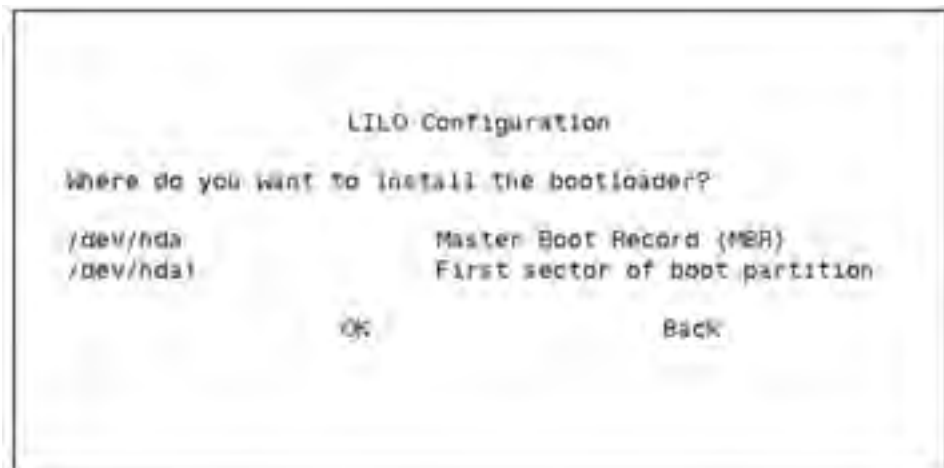
The first LILO configuration screen will appear, which asks about the use of linear mode. As it clearly says, it pertains to SCSI drives only—and we will address this issue in our SCSI installation after this section. On older versions of Red Hat distributions, it did not state that it was specific to SCSI installations. If you were to choose to enable this feature, it would have no effect—nor would it cause any problems. In Figure 3.8 we chose to remove the default `*`.

We plan to use LILO to manage the disk. Therefore, in Figure 3.9 we are going to choose to use the *Master Boot Record* (MBR) option. See Figure 3.10 for the resulting manager screen.

If there were some alternate operating system partitions on the disk, the LILO configuration would see them and you could choose to add them to your `/etc/lilo.conf` file as boot choices here. We only have the one system, and we will let this option default to boot Linux. We recommend that if this system is going to be a production system, no other operating systems should share the disk in order to primarily simplify the maintenance,



**Figure 3.8** LILO configuration: linear mode.

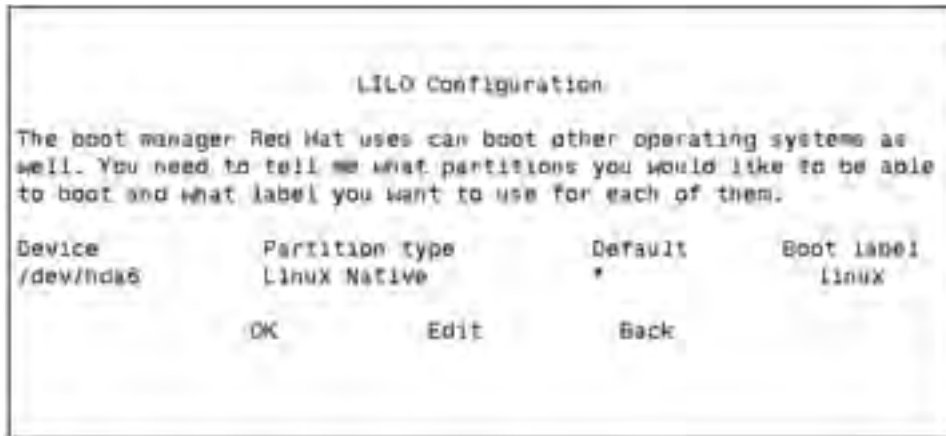


**Figure 3.9** LILO configuration: bootloader.

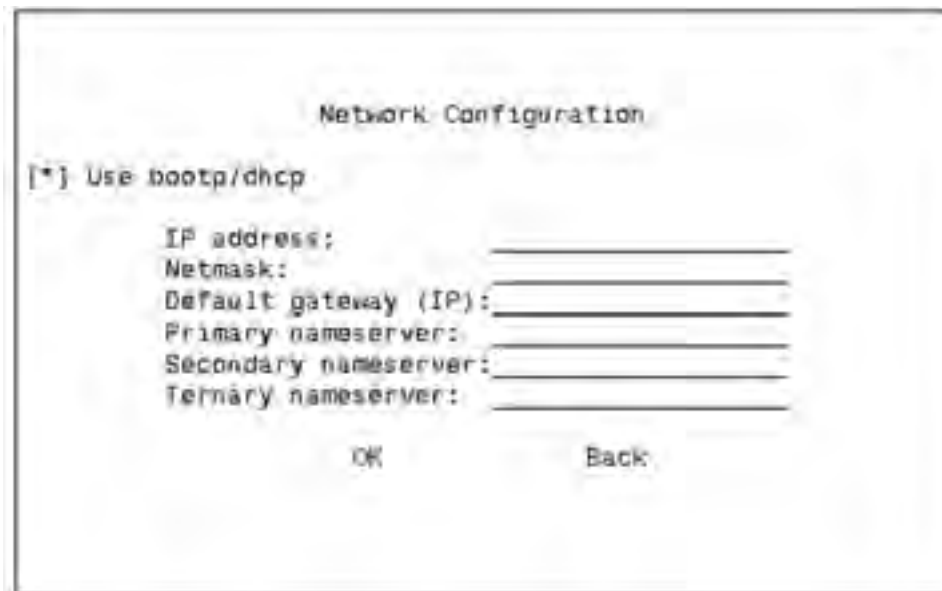
backup, and recovery of the system if need be. In other working environments, however, such as home, you might appreciate having more than one operating system from which to choose. See Figure 3.11.

### Step 8. Network Configuration

We are going to use a static IP address here. But if this system is going to be a print server or run any services for other hosts, it should receive its own static IP address so that it will not change. If the address changes, then



**Figure 3.10** LILO configuration: boot manager.



**Figure 3.11** Network configuration.

other hosts on the network might not be capable of finding the service. Another issue to examine is whether any software on this host is not sensitive to having the IP address changed. The IP addressing information we used was

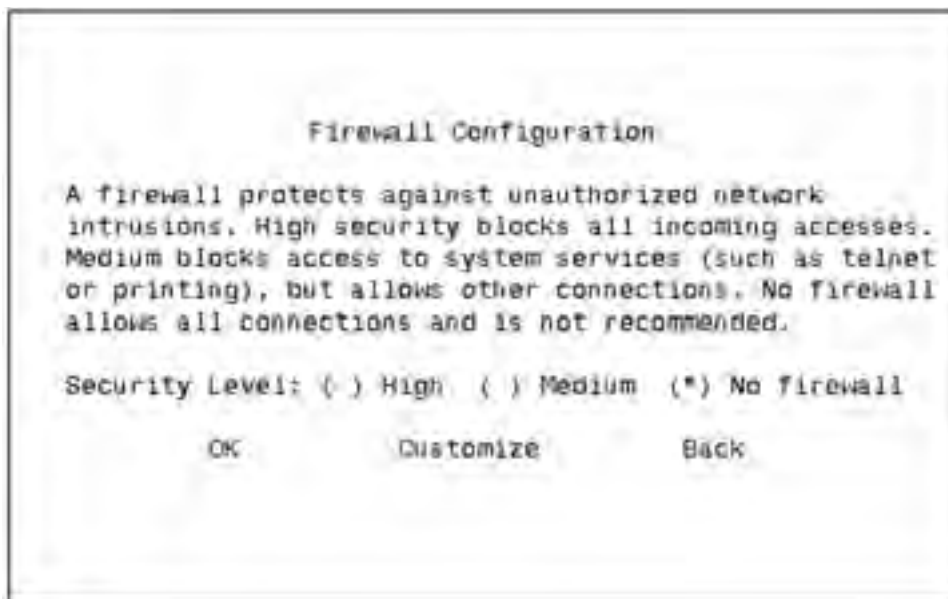
**DISABLE BOOTP/DHCP**

IP address:	192.168.2.205
Netmask:	255.255.255.0
Default gateway:	192.168.1.1
Primary nameserver:	24.64.2.33
Secondary nameserver:	24.64.2.34

You can revisit this menu at any time after the installation via `/usr/sbin/netconf` or `/usr/sbin/netcfg`. There is no opportunity during the installation to configure an ISP Internet connection. This setting can be configured after the installation by using `/usr/sbin/internet-config`. See Figure 3.12.

**Step 9. Firewall Configuration**

This function is a new feature added to the Red Hat 7.1 distribution. This choice will not appear on the Red Hat 7.0 distribution. This feature is in response to concerns that users had about security exposure on the Linux system itself. This particular feature set does not create a firewall in the sense that this system is a firewall for other systems. This feature set just



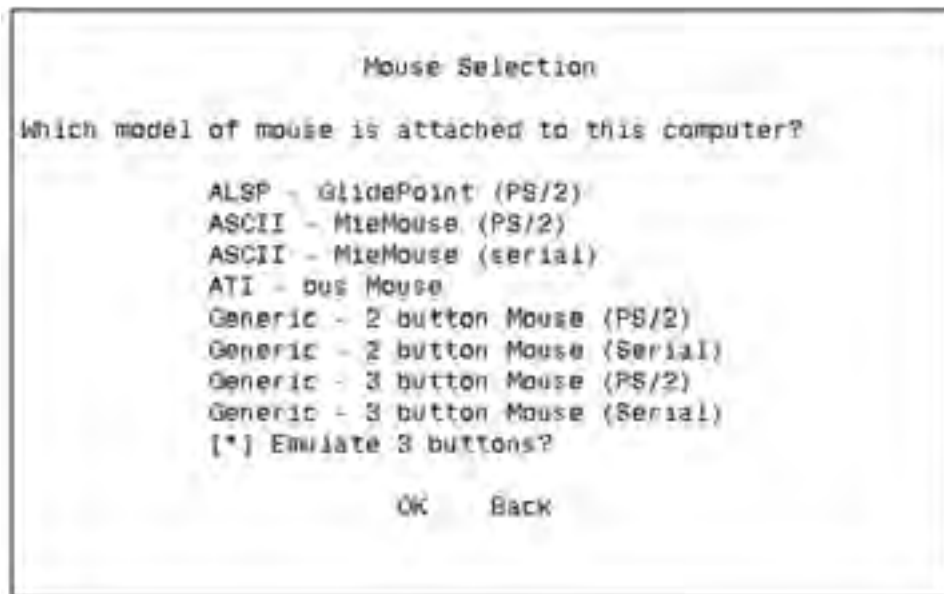
**Figure 3.12** Firewall configuration.



firewalls this system, per se. For example, if you choose Medium as an option, it will prevent Telnet access to the system and will not prevent Telnet connections through the system should it have two network cards in it. Linux can be a firewall, but this system is not that kind of configuration. We already have a proper firewall in place, and we wish to have easy Telnet access to this system from various points on the network so that we can manage it remotely; therefore, we are going to choose No firewall. If this system were to be a Linux system at the end of the ISP, we would recommend Medium or High so as not to be an easy target for hackers. You can reconfigure this information after the installation via `/usr/sbin/firewall-config` (see Figure 3.13).

### Step 10. Mouse Selection

The system immediately asks you to configure the mouse if it senses one. If you have a three-button mouse, you will be better equipped to take advantage of the X Windows system without confusion. Here is an example. On a three-button mouse, the left button is copy, the middle one is paste, and the right is used for context. For a two-button mouse, the left button is copy and the right is paste. Unless you choose to Emulate 3-buttons, then the left button is copy, the right is context, and the left-right buttons pressed together is equivalent to the middle button. We have a



**Figure 3.13** Mouse configuration.



**Figure 3.14** Language support.

two-button mouse. We chose Generic—two button Mouse (PS/2) and to Emulate three buttons. You can revisit this information after the installation via `/usr/sbin/mouseconfig`. Now, see Figure 3.14 for more information about language support.

### Step 11. Language Support

We chose English (USA) because we wish to have the normal code pages installed on this system. Installing more language support uses significant space on your disk. All message files, code pages, and related files such as screen fonts for X Windows will be installed along with the language support. If you are going to install additional or different language support here, give some thought to what you are going to use for keyboard support. Language support and keyboard support generally go hand in hand. This information can be reconfigured after the installation via `/usr/sbin/locale_config`.

### Step 12. Time Zone Configuration

Somewhere in your hardware setup plan, way back at the beginning, should have been instructions to set these elements properly in the system CMOS. Does your CMOS time function on your system handle GMT? If it does, and you are going to use the system itself as a source of time, then set

the time in the CMOS and use the hardware clock set to GMT option here. Then, it will convert the time set in your CMOS to the local time. If your CMOS date and time does not have the GMT capability and you set the time in your CMOS to the local time, then do not set the feature here. The decision to use a hard clock will ultimately have to do with whether or not this system will be a standalone entity as far as time goes. For example, will this system just be used by a developer or a system administrator as a utility system? If it were going to be a server, it might have to participate on a network using *Network Time Protocol* (NTP)—then it would be appropriate to make use of the Time Zone Selection. Plan for this situation ahead of time. We cannot tell you how messed up a database server can get if the time is set back in time. Not many database applications can recover from time changes gracefully. We chose not to use the hardware clock because we have the local time set in our CMOS and in Figure 3.15 we did use Canada/Mountain. This information can be reconfigured after the installation via `/usr/sbin/timeconfig`.

### Step 13. Root Password

Remember to choose a good password and then ensure that you remember it. Linux has rules and guidelines for proper passwords. Reviewing



**Figure 3.15** Time zone selection.

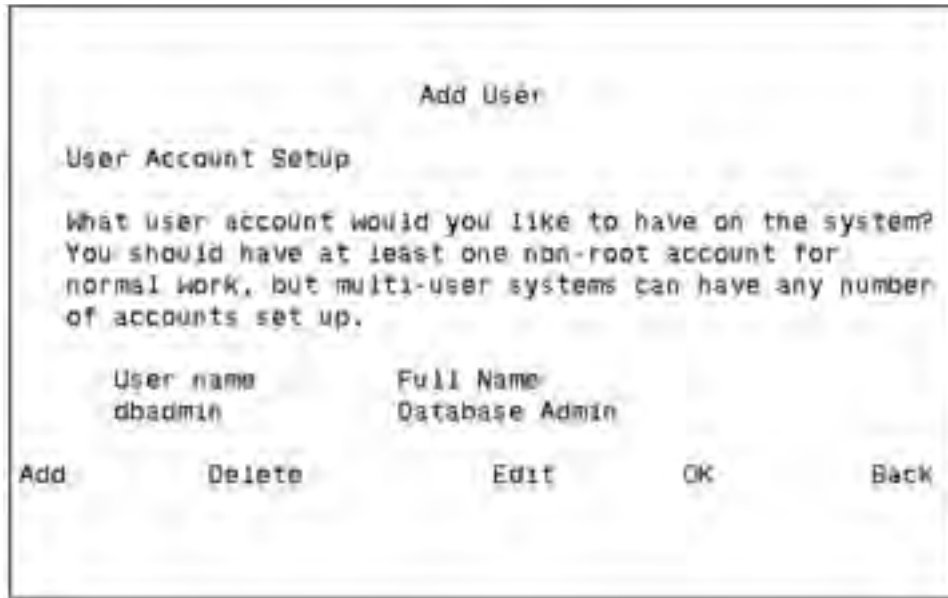
them would be worthwhile. Check the man pages and the Red Hat online documentation for guidelines. This password can be reset after the installation by first logging in as root and using the `passwd` utility. Set an appropriate password and enter it in your screen which is equivalent to Figure 3.16.

#### Step 14. Add Non Root Users

The installation process will enable you to add users at this time. You need not add all users now, though, because you can revisit this task at any time after completing the installation. The reason for the Add User dialog in Figure 3.17 is to allow you to log in to the system for the first time as a user and not as root. Technically, it is not proper form to log into the system as root. Therefore, for those who play only by the rules, the opportunity is here. We chose to create a `dbadmin` user at this time because we knew we were going to need it. We have plans to create all of our users after the installation is validated and ready for system administration. Another reason to create a non-root user is if you have only Telnet access to the system after installation. By default, the Telnet daemon will not enable you to log in directly as root. You have to log in as a regular user and then `su` to root. Therefore, if you only have Telnet access, immediately after the installation you will require this additional userid.



**Figure 3.16** Root password.



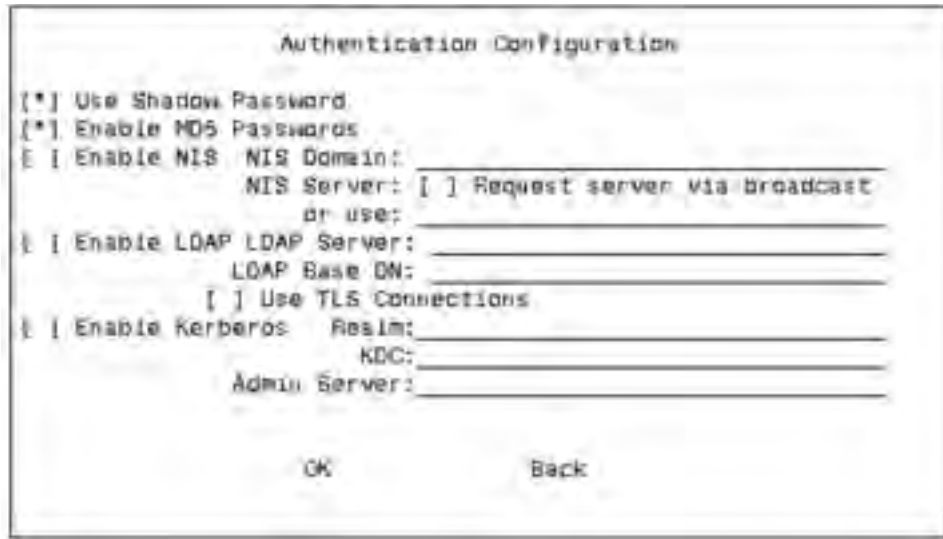
**Figure 3.17** Add a user.

### Step 15. Authentication Configuration

The two default selections here will ensure that you are compatible with other systems on the network. When you are configuring for *Network Information Services* (NIS), *Lightweight Directory Access Protocol* (LDAP) and Kerberos support is available for advanced configurations. These advanced configurations are almost mandatory in large, distributed server environments. These capabilities have always been available for Linux, but now they are configurable during the installation process. For Figure 3.18 we are going to use the defaults here: use Shadow Password and Enable MD5 Passwords in order to be compatible with our other operating systems on the network. This information can be reconfigured after the installation via `/usr/sbin/authconfig`.

### Step 16. Package Group Selection

Components are groups of software packages that provide an overall service or feature to the system. The components of our next screen represent our entries used for Figure 3.19. We did not display all the possible selections; rather, only the ones we suggest installing for this exercise. You might note that the dialog box on your screen is a little smaller than the one shown next; use the arrow key to reveal more components. We



**Figure 3.18** Authentication configuration.



**Figure 3.19** Package group selection.

installed these components. Support for additional features can be added or removed after the installation by using rpm, the Red Hat Package Manager. All of the associated services for various run levels can be managed

after the installation by using either `/usr/sbin/chkconfig` or `/usr/sbin/ntsysv` to modify the services:

- Printer support
- X Window System
- GNOME
- DOS/Windows connectivity
- Games
- Networked workstation
- NFS Server
- Kernel development
- Utilities

There is a potential here for frustrating errors. For example, if you had chosen to install a GNOME workstation, these packages would be different. Although you would get networking capability, you might not have Telnet or FTP functionality (which you might think is important). Normally, you would only get Telnet, FTP, and NFS if you had chosen Server. Now is your chance to add all this functionality by choosing to Select individual packages. It is much easier to add this functionality here than to install and configure them later after the installation.

After having worked with earlier versions of Red Hat, we were used to working with utilities like `linuxconf` and `XFree86Config`. We consider these packages an absolute must for the system administrator. Do not trust the Utilities function. Go into the Select individual packages under Applications/System and User Interface/X and go get them. While you are there, check for all your Telnet, FTP, and other essential daemons, as well. Some daemons that were standard fare are not anymore with the newer distributions. Although they can be added afterward, it is easier to deal with them here.

If you scroll down through these dialog boxes, you will eventually see the Everything choice at the bottom of the component list. We mentioned earlier that choosing to install everything will result in 2GB of files and directories on your system. If you are performance minded, you might want to be somewhat selective at this stage. If performance is not a primary configuration, you might want to choose all components (and thus, all of the packages). See Figure 3.20.



**Figure 3.20** Package requirements.

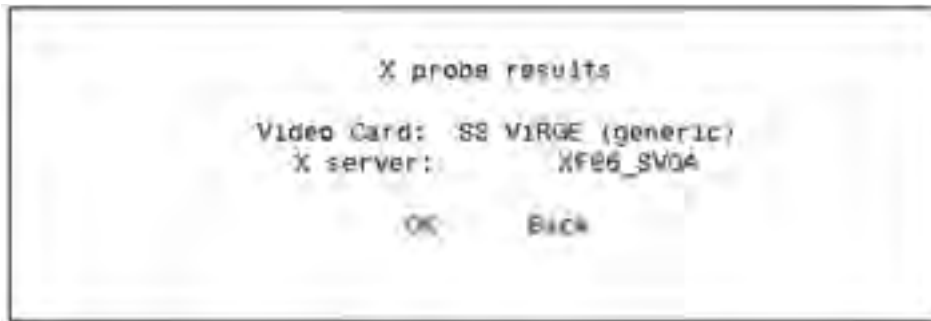
**WARNING** The installation program will tell you whether you have inadequate disk space. In the meantime, do not forget your planning and do not make disk space too tight in the / and /usr file system partitions. Finally, be sure to avoid removing mandatory packages, such as kernel support and certain libraries and their respective requisites.

If you choose to install a particular package that uses another package as a prerequisite, the installation process will bring that to your attention and will install all required software for you. We recommend choosing the Install packages to satisfy dependencies option.

### Step 17. Video Configuration

In our case, the installation process has conducted a probe by using the PCI Probe and found the video adapter. We recommend at this point that you make a note of the video card and X server response and move along. Our experience is that it will fail on most systems most of the time. We have an entire chapter dedicated to configuring the video on a system. Remain focused on other features and functionality at this point, and move through the video configuration—accepting all of the suggested defaults.





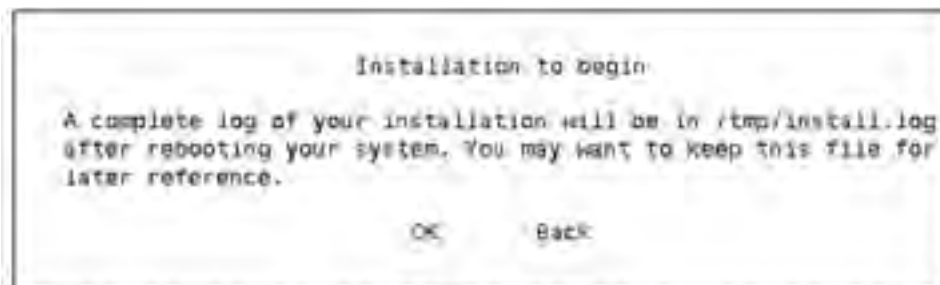
**Figure 3.21** X Probe results.

We were having a good day, and it recognized our video chip set. The video configuration can be reconfigured after the installation via `/usr/bin/X11/Xconfigurator`. For Figure 3.21 we simply chose OK to continue with the installation to move on. See Figure 3.22.

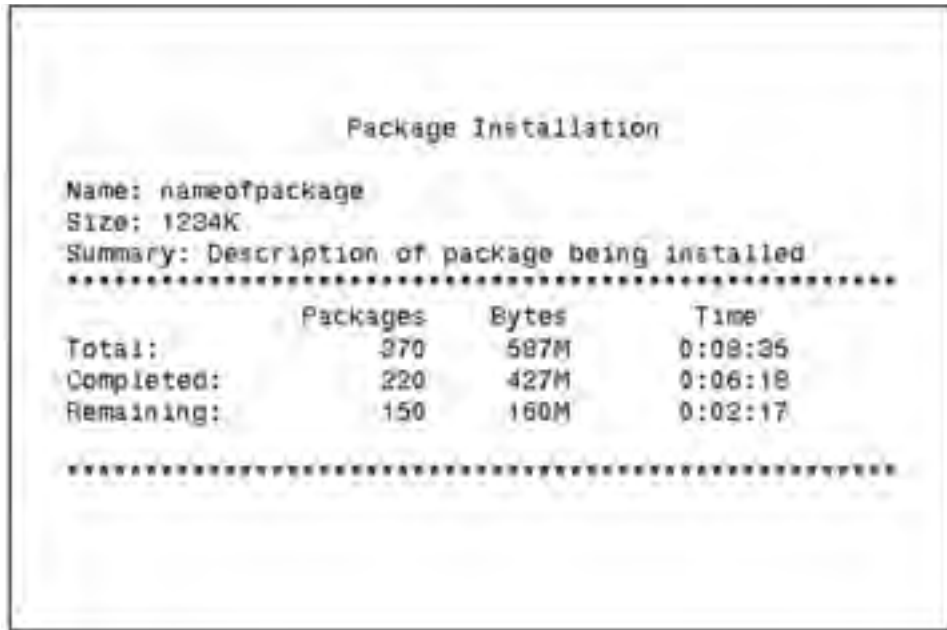
### Step 18. Installation

A log file called `/tmp/install.log` will remain after the installation, and you can use it to check and ensure what files and packages were installed. This log can also serve to be good documentation in the event that you ever have to recreate or recover a system without a backup system. See Figure 3.23 for the package installation screen.

A process screen will keep you informed as to the status of the file copy process. Throughout the installation process, you will see this installation status dialog box, which will show you the progress of the installation of



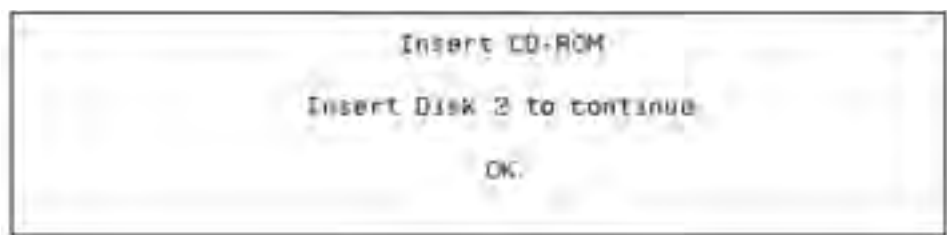
**Figure 3.22** Install to begin.



**Figure 3.23** Package installation.

individual packages and programs as well as the overall progress of the total Linux installation. Note in the top half that as the individual packages install, the installation program will provide the package name, the size of the package, and a brief description of the package.

On the bottom, the program keeps track of the space used by the install packages as well as the remaining space required. Moreover, it tracks the total, elapsed, and remaining installation time. This point might be a good time for you to take a break while the installation proceeds. See Figure 3.24.



**Figure 3.24** Insert the CD-ROM, disk 2.

If the second CD is required, it will be requested. Do not worry if the second CD is not required; it just means that you did not choose to install any packages from that CD. If the CD does not mount immediately, give it some time and try again.

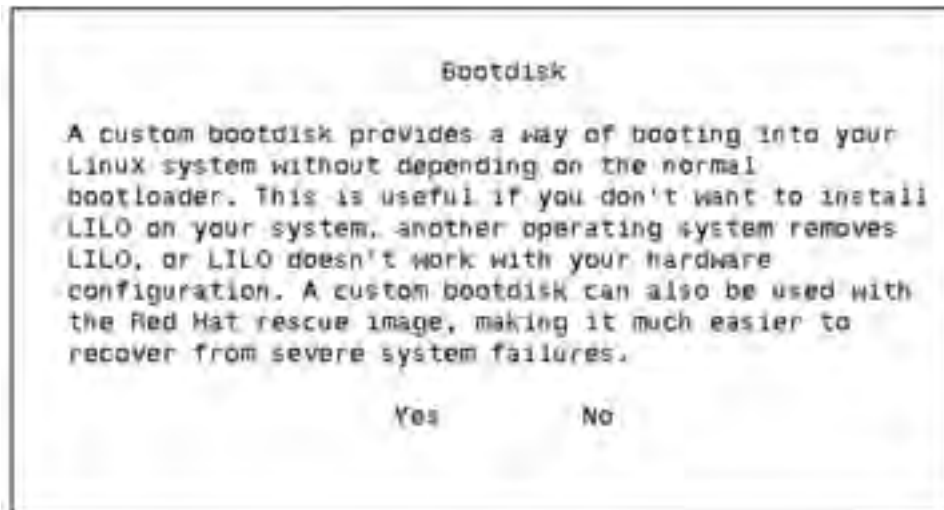
### Step 19. Create a Custom Bootdisk

Respond to Figure 3.25 by inserting a blank floppy disk and then pressing OK (see Figure 3.26).

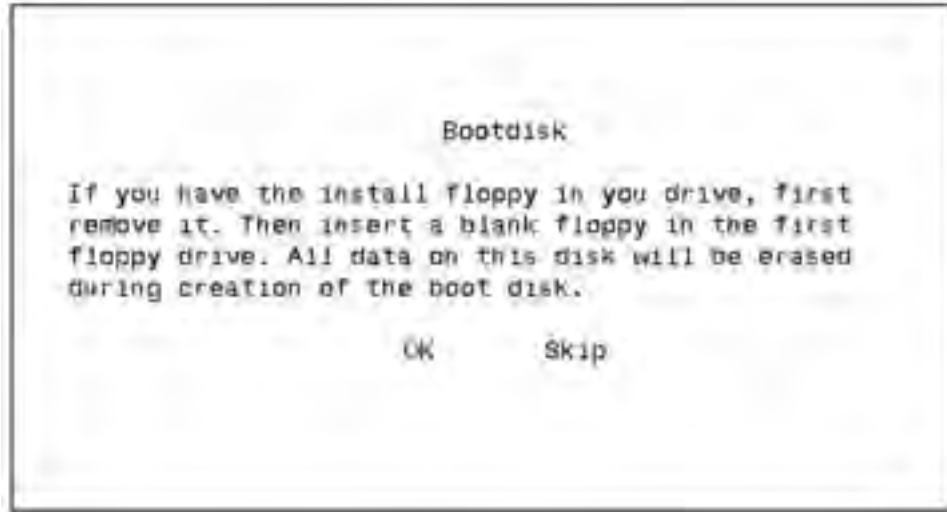
Although you can create a custom boot disk at any time, we recommend that you make one now. Did you know that these rescue disks are not generic but are specific to each system? Do it. Create a label for the disk while you wait for the procedure in Figure 3.27 to complete. See Figure 3.28 for information about monitor setup.

### Step 20. Monitor Setup

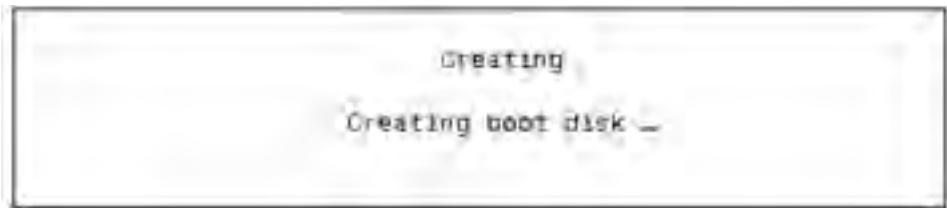
Although scores of monitor names appear in the next screen, your monitor might not show up in the list. If it does not, you can choose Custom here. If you choose Custom, it will try to test the configuration later. In the event your video fails, do not worry and move on. We will show you how to configure the monitor when we show you how to configure the video in another chapter. Again, we were having a good day; the Gateway Vivitron 17 was in the list.



**Figure 3.25** Bootdisk request.



**Figure 3.26** Bootdisk—Warning or Skip



**Figure 3.27** Bootdisk—Creating

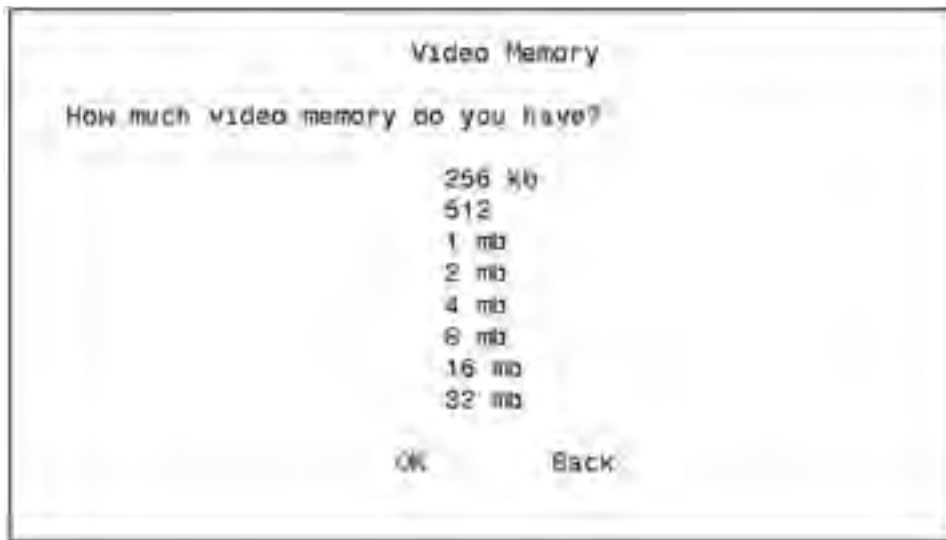
### Step 21. Back to Video Configuration

At this point, our installation might not reflect what your installation process is doing because these menus are dynamic with respect to previous responses made. In our case, it probed and located a video card and is now prompting us to indicate how much memory is available on the video card. Again, we emphasize that you should move on—and we will reconfigure the video and monitor later using a better utility. Our video card has 4MB of memory on it. For Figure 3.29 we chose 4MB. Choose what is correct for your video card if you know it.

The only way that you would know which clockchip you had was to read the documentation that came with the video card. We are choosing to configure the video later, so for Figure 3.30 we chose No Clockchip Setting and moved on. Once the system is running, you can run `/usr/bin/X11/SuperProbe` to identify the video chipset.

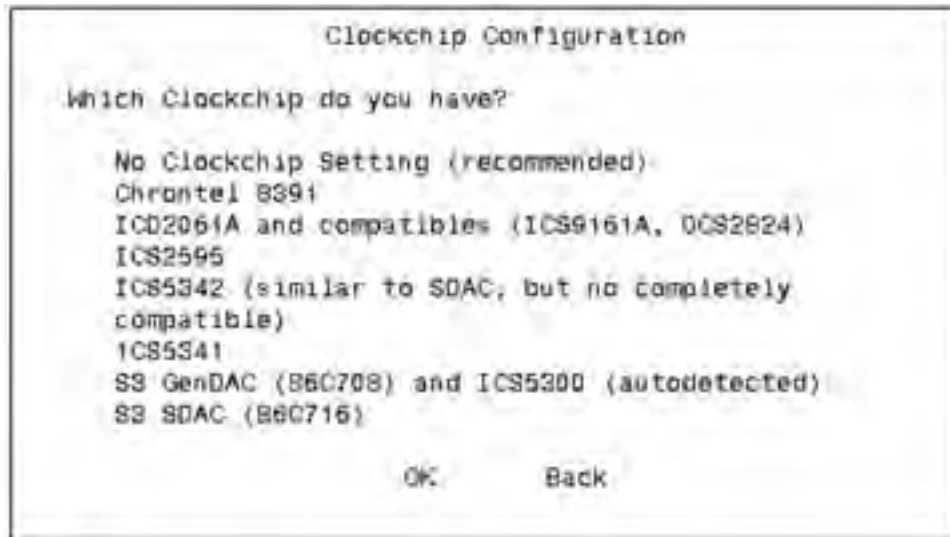


**Figure 3.28** Monitor setup.

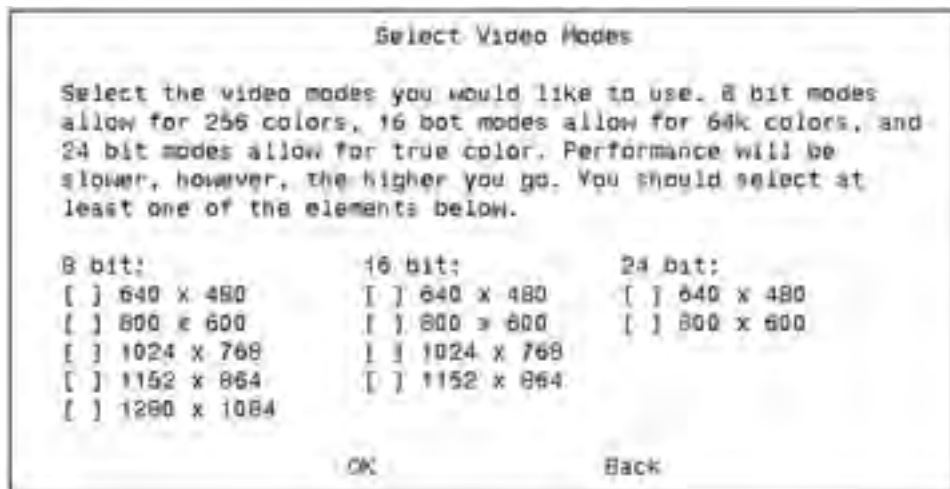


**Figure 3.29** Video memory.

Although we are going to configure the video setting later, you have to choose at least one here. We recommend that if you choose anything for Figure 3.31, choose at least these: 8 bit:  $800 \times 600$ ,  $1024 \times 768$ , 16 bit:  $800 \times 600$ , and  $1024 \times 768$ . These are representative of the lowest common denominator (what most monitors can handle).



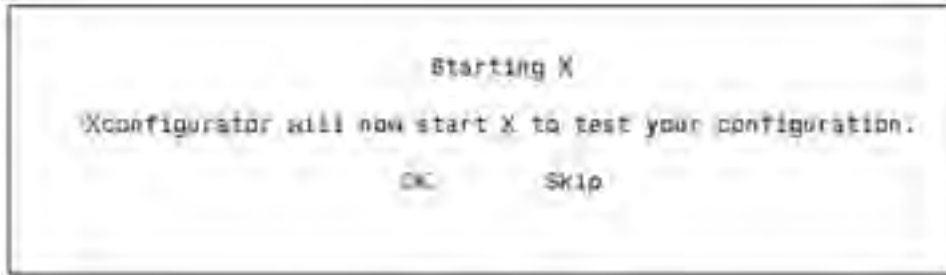
**Figure 3.30** Clockchip configuration.



**Figure 3.31** Select video modes.

## Step 22. X Window Configuration

You can choose **OK** here just for fun at this point, but if your video is not going well, this option will definitely not work. If it works, then choose **Yes**. Only choose **Yes** if you do not see any irregularities. A typical irregularity would be a box or small images hanging around the mouse icon or black lines on the screen where they should not be. Respond **No** if there



**Figure 3.32** Starting X.

are any irregularities. We chose NO for Figure 3.32. When and if we need the GUI interface we will start X manually, otherwise the server will run without leaving the X in memory. Saves resources!

**WARNING** You might also be presented with a screen asking you whether you would like the X window desktop to automatically start when the system starts. We highly recommend that you say No in case the desktop hangs up the system. You can always start the X window desktop at any time by entering `startx` at the command line after you log into the system.

### Step 23. Complete

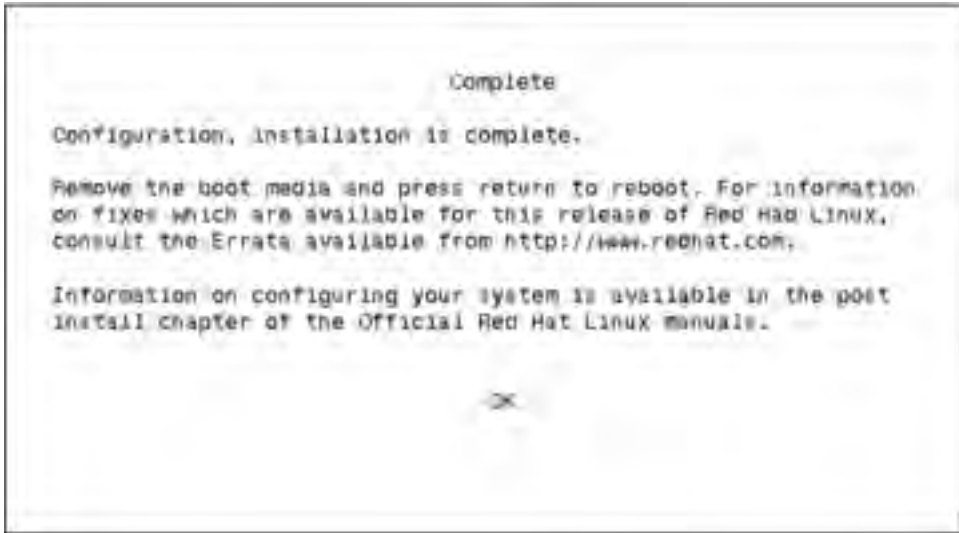
At this point, you should reboot the system and test the various functions and features that you have selected (see Figure 3.33).

## Installing Linux on a SCSI System

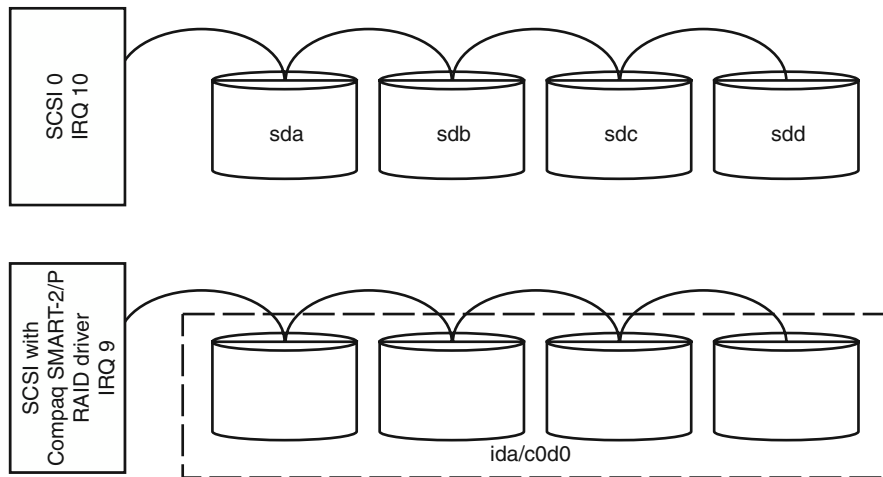
---

It is our hope that although most of the installation screens are the same, most of the conversation will be significantly different when we reach the disk partitioning and mount point screens. This section will enable us to demonstrate some manual partitioning techniques by using a SCSI drive system. Our SCSI drive system also includes a RAID array. The disk configuration used for this installation example is as follows (see Figure 3.34).

Our goal is to build a server that is suitable for a small e-service. This server would include a Web server and commerce software with a data-



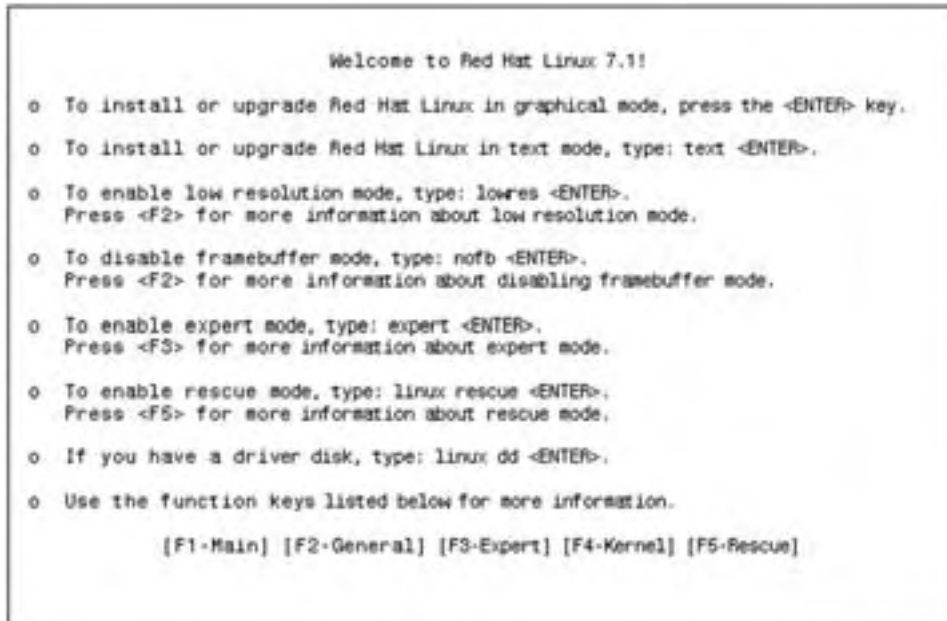
**Figure 3.33** Complete.



**Figure 3.34** SCSI disk configuration.

base manager. Steps 1 through 5 are identical to the IDE example installation; therefore, the discussion begins at Step 6 that deals with manual partitioning over the SCSI RAID array. See Figure 3.35.





**Figure 3.35** Welcome To Red Hat Linux 7.1! (boot message).

### Step 1. Boot Message

Choose text to get to the next screen.

### Step 2. Select Language

Among the choices shown in Figure 3.36, choose English to get to the next screen.

### Step 3. Select Keyboard

Among the choices shown in Figure 3.37, choose us (United States) to get to the next screen.

### Step 4. Welcome

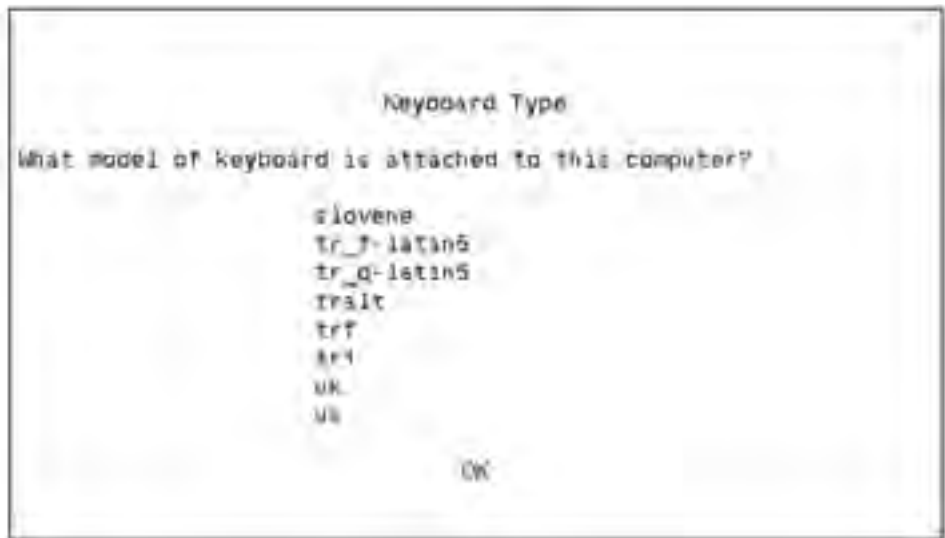
Once we reach the installation screen as shown in Figure 3.38, choose OK to get to the next screen as shown in Figure 3.39.

### Step 5. Specify Installation Type

Here, we chose Custom so that we could assemble a specific set of packages to support our software and system management environment.



**Figure 3.36** Select language.



**Figure 3.37** Select keyboard.

### Step 6. Partitioning

In Figure 3.40, we chose to Manually partition our drives because we have a very specific configuration in mind. You will also notice that if we did



**Figure 3.38** Welcome to Red Hat Linux! (installation).

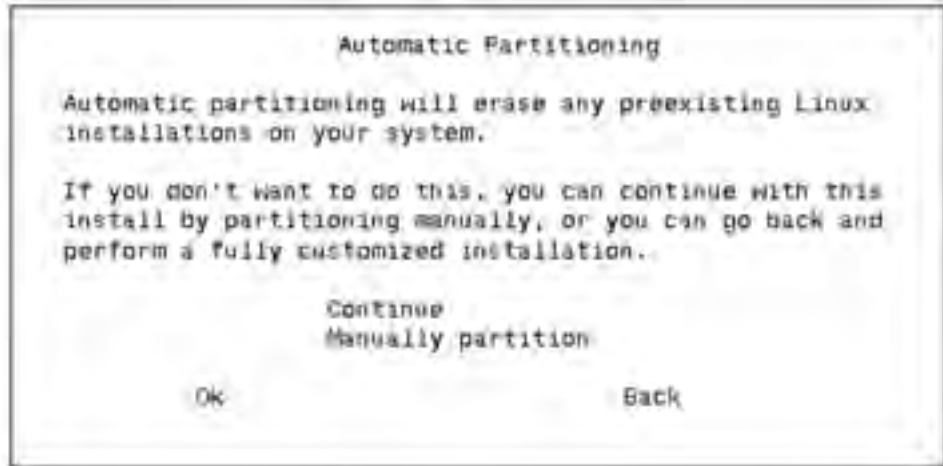


**Figure 3.39** Installation type.

not intervene, we would have ended up with a very strange situation because the default behavior would have yielded some pretty weird results. We will run through those details with you in the next set of screens. Figure 3.41 shows the Disk setup screen.

We are going to use Disk Druid to create the new partitions and mount points required. See Figure 3.42.

This display is what our first screen looks like. If you were using drives that had some kind of previous data or operating system, you might have



**Figure 3.40** Partitioning.



**Figure 3.41** Disk setup.

received some errors about not recognizing the current partition(s) or not being able to read the existing partition tables. If this situation happens, we recommend that you just hit Enter to go past all the warnings and errors

Current Disk Partitions				
Mount Point	Device	Requested	Actual	Type
Drive Summaries				
Drive	Geom [C/H/S]	Total	Used	Free
sda	[ 8676/ 64/32]	8676M	0M	8676M
ida/c0d0	[ 6633/285/32]	26029M	0M	26029M
sdb	[ 8678/ 64/32]	8678M	0M	8678M
sdc	[ 8678/ 64/32]	8678M	0M	8678M
Add Edit Delete Ok Back				

**Figure 3.42** Current disk partition—none.

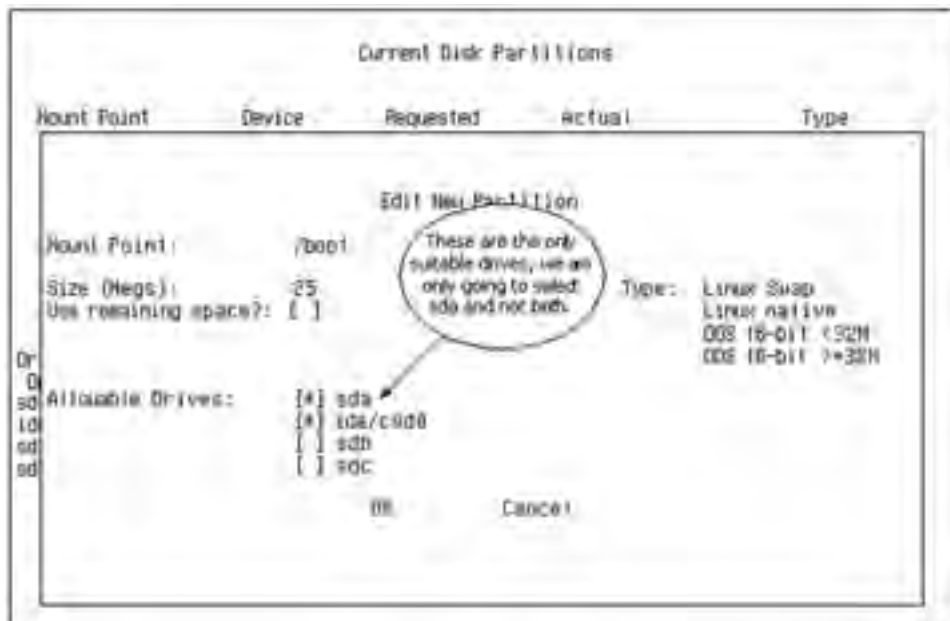
until you reach this screen. When you get to this screen, you will have an opportunity to remove all the existing partitions by using Disk Druid. If you can still see some residual, older partitions, you can delete them here. Highlight the old partition and use Delete. *This action will destroy the data.*

If you cannot get to this screen because of the errors, you might have to run some low-level disk utilities on the drives to remove any existing partitions or previously encoded data. Please take our advice—if you are going to use low-level utilities on these kinds of expensive drives, contact the manufacturer of the drives to obtain the utilities. Do not let anyone talk you into using his or her configuration tool for this purpose until you know that the manufacturer will support it.

First, let's interpret the Drive Summaries information before we proceed. We see four drive objects called sda, ida/c0d0, sdb, and sdc. sda is SCSI drive number one. sdb is SCSI drive number 2. sdc is SCSI drive number 3. You cannot see sdd, which is SCSI drive number 4, because it contains a hidden Compaq partition with some special network utilities on it. We have chosen not to make it available for this configuration. ida/c0d0 represents the RAID subsystem. The RAID subsystem contains the other four drives.

Our plan is to create separate partitions/mount points called /boot, /usr, /, /tmp, and /var on a non-RAID drive called sda. We are going to put the swap partition on its own drive, called sdb. All user data is destined for the RAID drive: /home, /www\_root, /db\_volume1, and /db\_volume2. To perform this task, we select Add to create a new partition (see Figure 3.43).

This menu appears. At the mount point, enter /boot. In the Size (Megs), enter 25. Leave the type as Linux native. Now, you have to be very careful with respect to how you interpret the Allowable drives. It is using the \* symbol to indicate which of all the drives are valid choices for the /boot mount point. Remember that placing the boot files in a disk area that straddles or exceeds cylinder 1023 could cause problems. You do not have to choose both. As a fact, you should not leave both as choices. Remove the choice under ida/c0d0 and leave only one \* at the sda, as illustrated in the figure. We do not want to put the /boot mount point on the RAID drives. It is our configuration choice to separate the operating system files and directories from the user and application data whenever possible. This functionality will enable us to move the user data, Web content, and database data to a different system sometime in the future should we decide to upgrade the system. Mixing operating system and user data would make

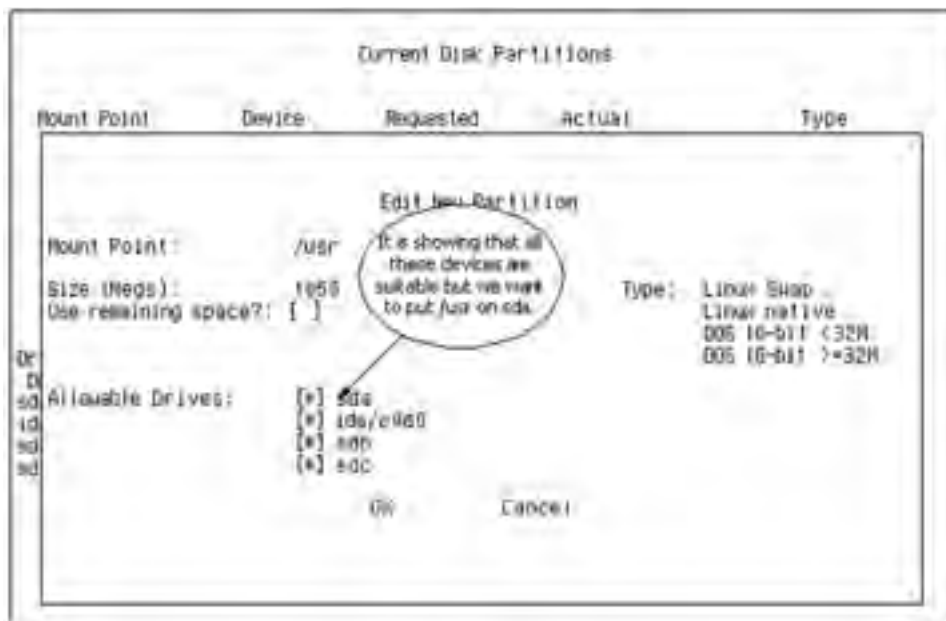


**Figure 3.43** Current disk partition—/boot.

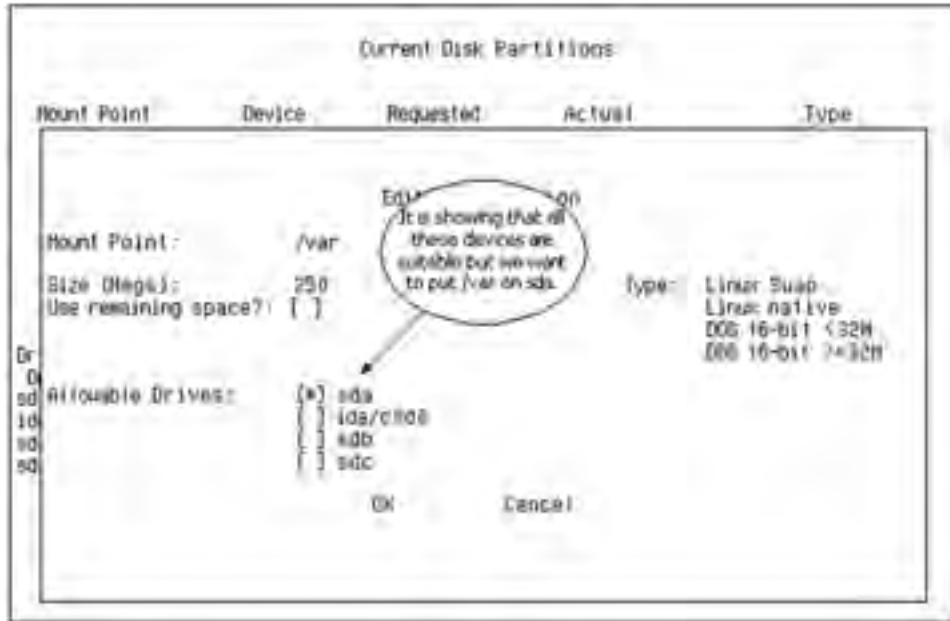
this process difficult. One could argue that in this configuration, if the sda drive fails, then all is lost. While this statement is true, we still have a free drive (sdc) that we could configure as a mirror. If we were to perform this action, we would then decide to move the swap partition back onto sda and then mirror the entire sda drive to the sdc drive. See Figure 3.44.

The next mount point is /usr. At the mount point, enter /usr. In the Size (Megs), enter 1060. It was decided to reserve the /usr mount point for operating system applications and utilities. When we install the Web and database software, we will create new mount points for each one, respectively. Leave the type as Linux native. Again, you have to be very careful with respect to how you interpret the Allowable drives. It uses the \* symbol to indicate which of all the drives are valid choices for the /usr mount point. In keeping with our plan, we are going to remove all choices except the one \* at the sda. From this point on, we are going to show the modified version of the screens. See Figure 3.45.

/var is typically where all of the operating system and applications logs accumulate. Based on experience, we have found that 250MB is enough. We have a set of system administration scripts including /usr/sbin/logrotate that run constantly to monitor the size of all logs in /var. These scripts also perform rudimentary error log maintenance if the application



**Figure 3.44** Current disk partition—/usr.



**Figure 3.45** Current disk partition—/var.

does not provide such facilities. In either case, if the error logs fill this space, the other spaces are not affected. See Figure 3.46.

Based on our experience, /tmp is very unpredictable. Many applications and utilities use this space as a work area. For this reason, we like to create a separate mount point for it, too. Other permission and security issues arise over the use of /tmp, as well. Some application software requires users to have access to the /tmp directory to provide space for temporary application files. If this situation were the case, we would prefer to allow access to a separate area rather than a directory that is aligned with the actual operating system.

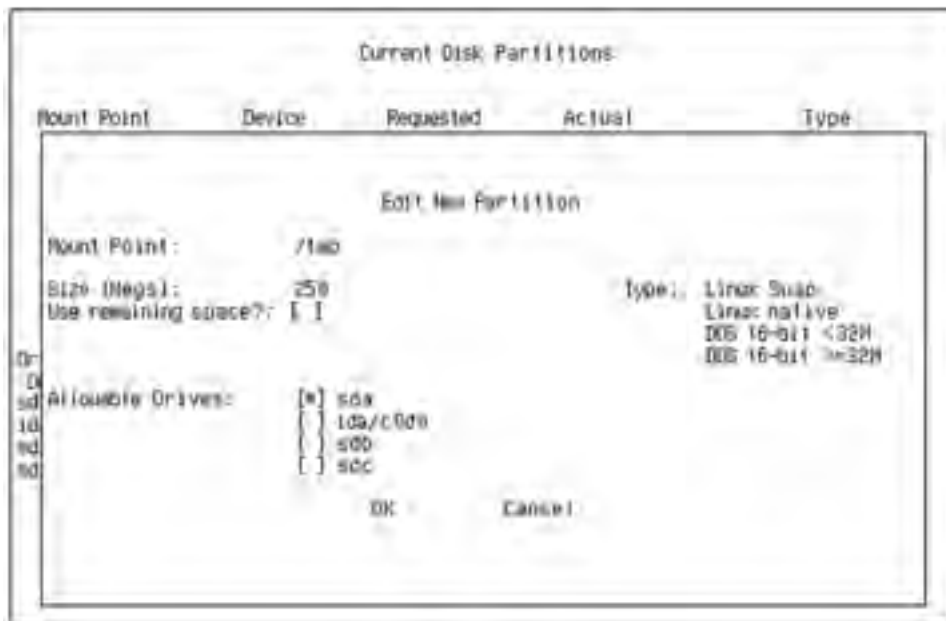
All other directories will end up in / (see Figure 3.47). At this point, however, we have taken care of most of our issues.

Although the /home mount point as shown in Figure 3.48 is typically created as part of the operating system, it is a place for user data; therefore, it is being directed to the ida/c0d0 drive.

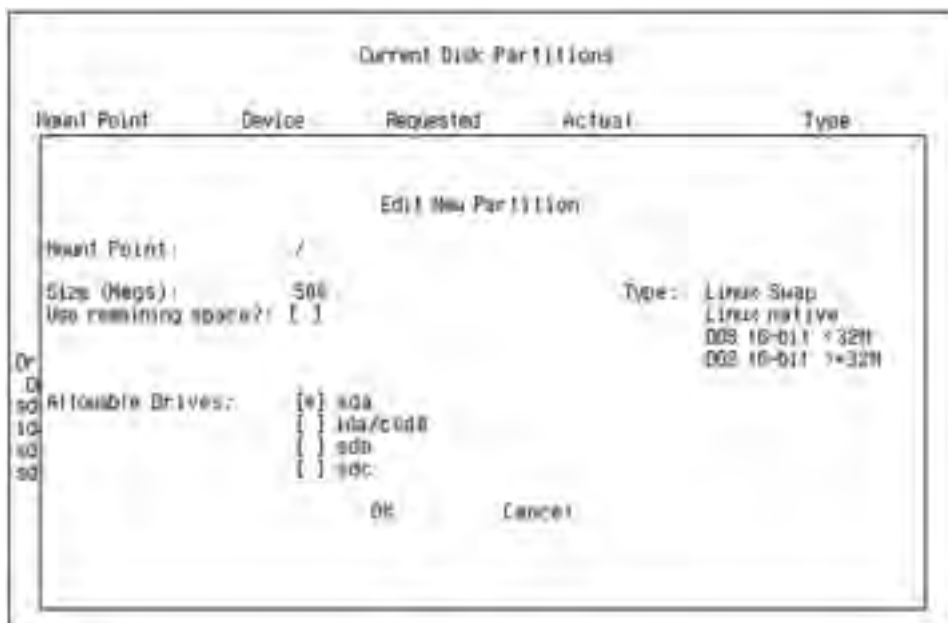
This mount point is our first user-defined mount point, called /db\_volume1 (see Figure 3.49). This volume was created to hold database files. It will go on the RAID system, as well.

A second user-defined mount point called /db\_volume2 will be created to hold an alternate database product (see Figure 3.50). This database will

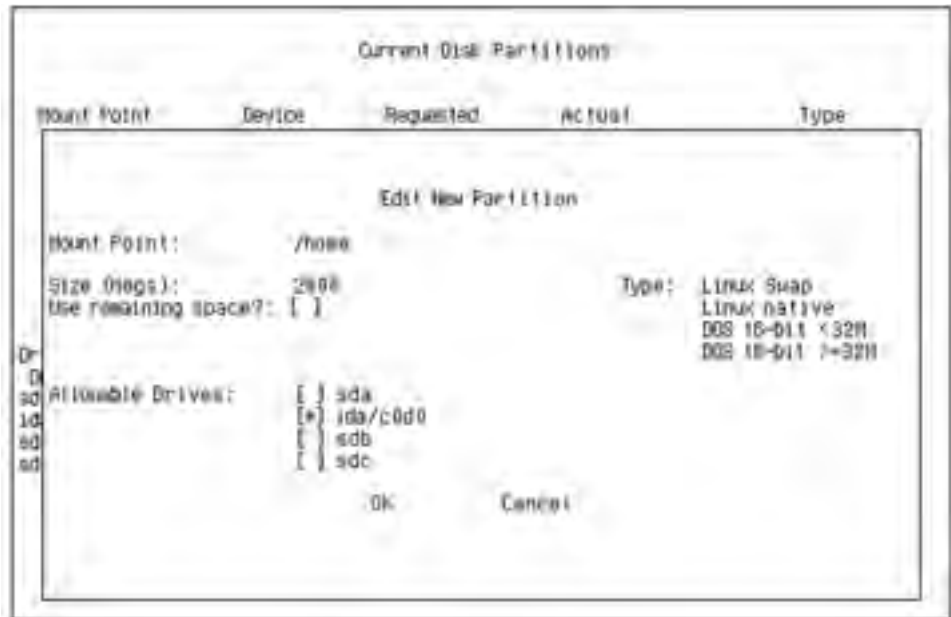




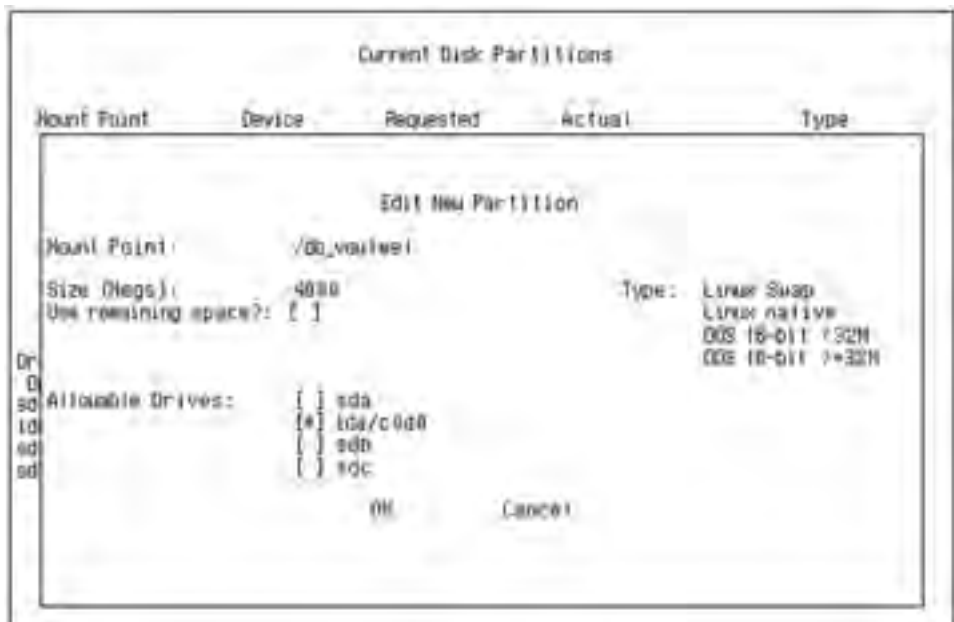
**Figure 3.46** Current disk partition—/tmp.



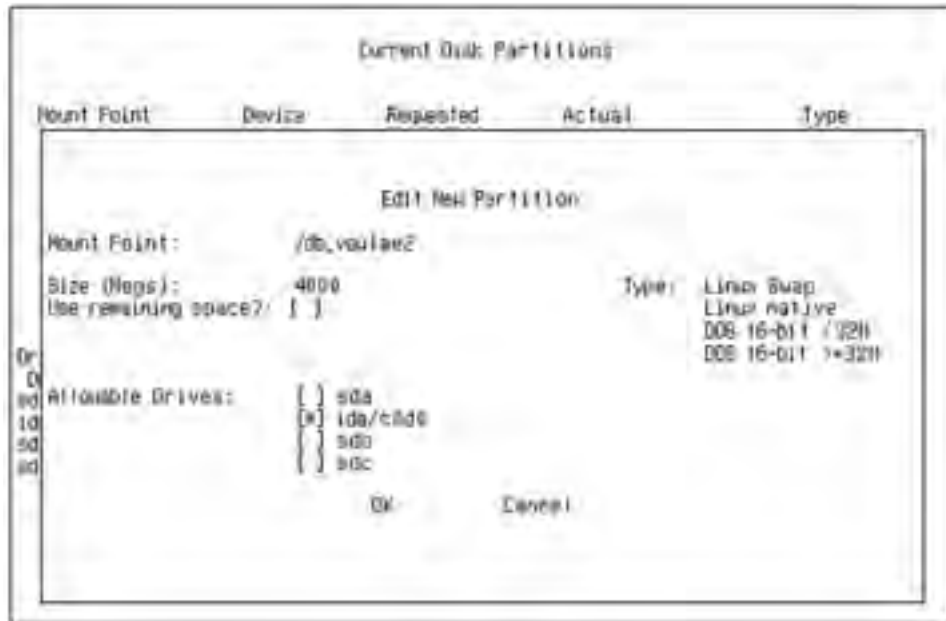
**Figure 3.47** Current disk partition: /.



**Figure 3.48** Current disk partition—/home.



**Figure 3.49** Current disk partition—/db\_volume1.



**Figure 3.50** Current disk partition—/db\_volume2.

have a different backup schedule than the first. It is part of our plan to keep all related database project information together in their own partitions to facilitate our backup and restore requirements.

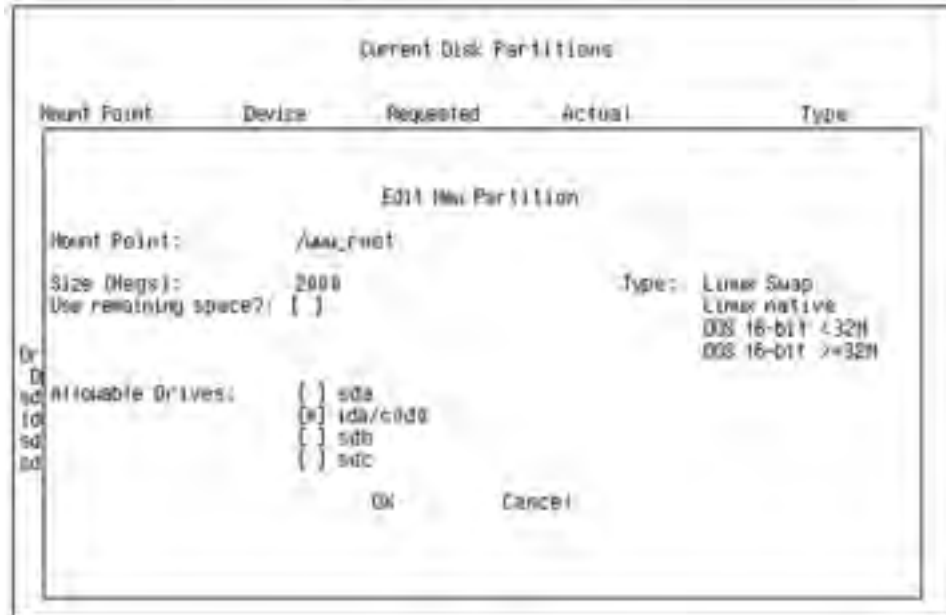
This system is also going to be a Web server. A /www\_root mount point was created to keep the Web content together for security and backup reasons (see Figure 3.51).

Try to place the swap partition on a drive that will have little or no other activity on it (see Figure 3.52). To place a swap partition on a drive that has an active database or file is poor practice. This feature will not only help with performance but will also enable us to increase the size of the swap partition by using contiguous blocks on the disk.

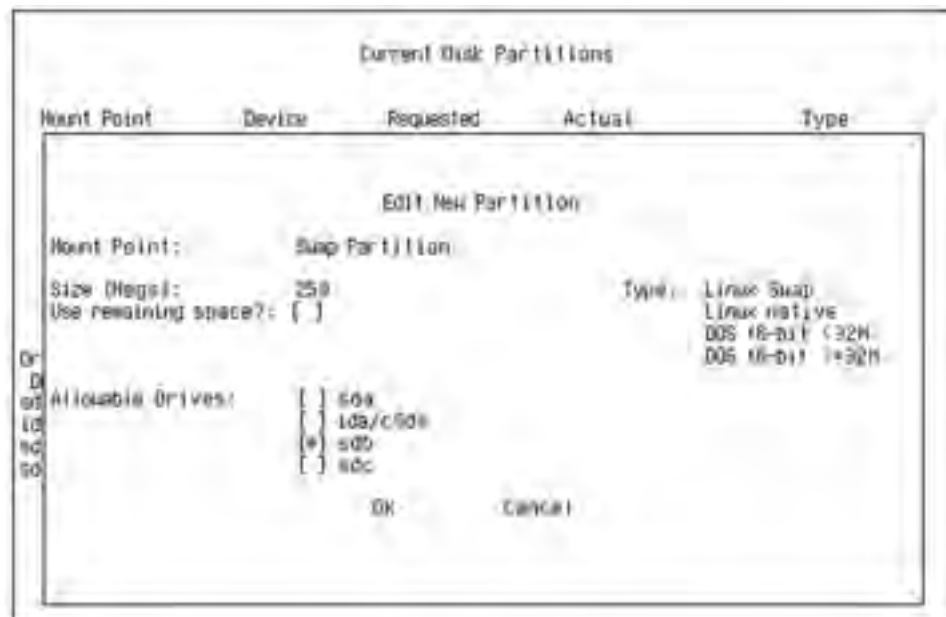
This list is our final list of disk partitions (see Figure 3.53). The logical naming of the partitions and mount points is as described in the section of IDE drives. Sdx (and not hdx) denotes the main differences between SCSI drives.

This point is where the partition tables are actually modified. No partition information is written to the disk until now. At this time, we indicated Yes to Save changes to partition table(s) as shown in Figure 3.54.

Another screen appears that asks you whether you would like to format the partitions (see Figure 3.55). This formatting is destructive and will remove or destroy any data on the partition if one existed previously. This



**Figure 3.51** Current disk partition—www\_root.



**Figure 3.52** Current disk partition—swap partition.

Current Disk Partitions

Mount Point	Device	Requested	Actual	Type
/boot	sda1	50M	50M	LINUX native
/usr	sda5	1000M	1000M	LINUX native
/	sda6	500M	500M	LINUX native
/tmp	sda7	250M	250M	LINUX native
/var	sda8	250M	250M	LINUX native
/db_volume1	ida/cdd0p1	4000M	4000M	LINUX native
/db_volume2	ida/cdd0p5	4000M	4000M	LINUX native
/home	ida/cdd0p6	2000M	2000M	LINUX native
/www_root	ida/cdd0p7	2000M	2000M	LINUX native
	sdb1	250M	250M	LINUX swap

Drive Summaries

Drive	Geom (C/H/S)	Total	Used	Free	
sda	8678/ 64/32	8678M	2994M	6684M	##
ida/cdd0	6585/265/32	26929M	12000M	14929M	####
sdb	8678/ 64/32	8678M	250M	8428M	
sdc	8678/ 64/32	8678M	0M	8678M	

Add      Edit      Delete      On      Back

Figure 3.53 Current disk partitions—final construct.

Current Disk Partitions

Mount Point	Device	Requested	Actual	Type
/boot	sda1	50M	50M	LINUX native
/usr	sda5	1000M	1000M	LINUX native
/	sda6	500M	500M	LINUX native
/tmp	sda7	250M	250M	LINUX native
/var				LINUX native
/db_volume1				LINUX native
/db_volume2				LINUX native
/home				LINUX native
/www_root				LINUX native

Drive Summaries

Drive	Geom (C/H/S)	Total	Used	Free	
sda	8678/ 64/32	8678M	0M	8678M	

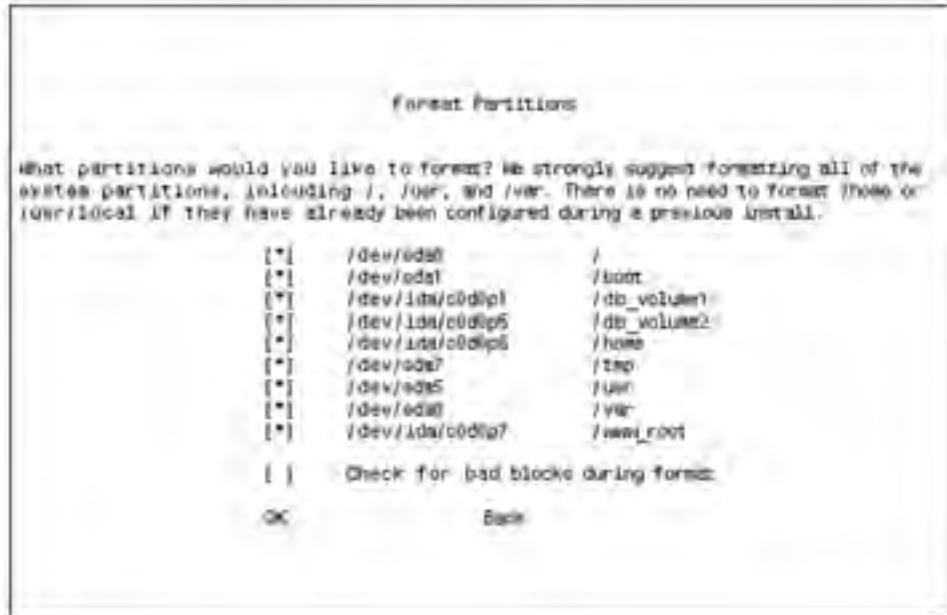
Add      Edit      Delete      On      Back

Save Changes

Save changes to partition table(s)?

Yes   No   Cancel

Figure 3.54 Save changes to partition table(s).

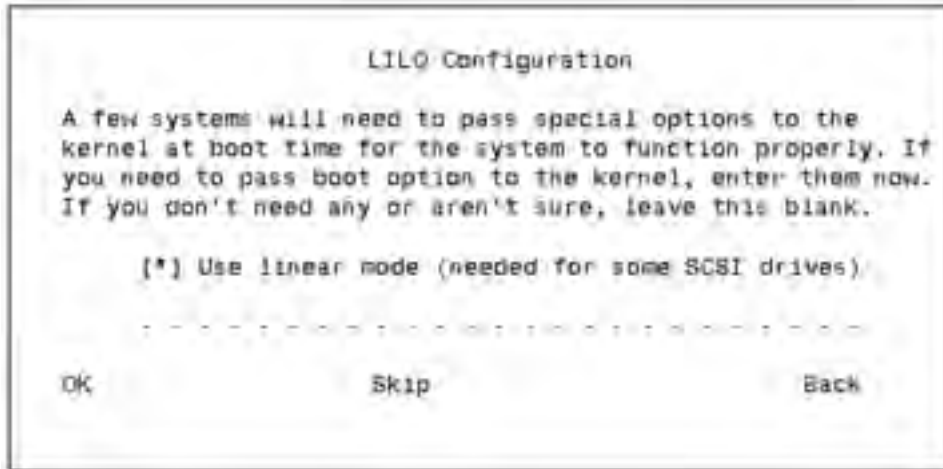


**Figure 3.55** Format partitions.

format is a high-level format and runs quickly, whereas if you choose Check for bad blocks during format, this process could take more than four hours depending on the size of the partitions. This format is a low-level format and runs slowly. The description also mentions that there is no need to format /home or /usr/local partitions. Sometimes this tool is used post-installation to add or modify partitions on drives. When done, it is sometimes desirable to retain the existing information in both the /home and /usr/local partitions. In other words, if you repartition an area that already exists without adjusting the size or changing the mount point, then Disk Druid will maintain the information in the partition (in which case, you would not want to format that partition). See Figure 3.56.

### Step 7. Installing and Configuring LILO

Now that the installation process knows where the /boot partition is going to be located, it has enough information to construct the `/etc/lilo.conf` file. The first thing that it requires is access to the drive that has the boot files. Sometimes, a Linux system will hang up upon booting and display a partial prompt. For example, you will see `LI` and not the complete `LILO:` prompt. Linux cannot locate the boot information in this case because it is



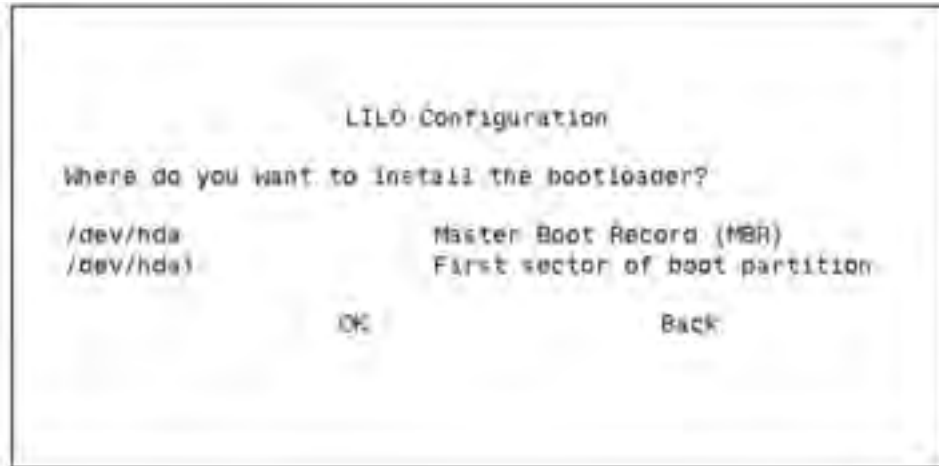
**Figure 3.56** LILO configuration: linear mode.

not correctly using the disk geometry supplied by the BIOS. Enabling this option will cause LILO to use *linear block address* (LBA) mode instead of the default CHS mode to locate the boot files. This situation will add an additional line with the keyword `linear` to the `/etc/lilo.conf` file. There is an example of this file on the next page. If you are already installed, you do not have to rerun the installation panels. You can modify the `/etc/lilo.conf` file directory and reboot the system. Now, because the system will not boot from the hard disk, you have to use the rescue floppy that you created to get the system up and running. See Figure 3.57

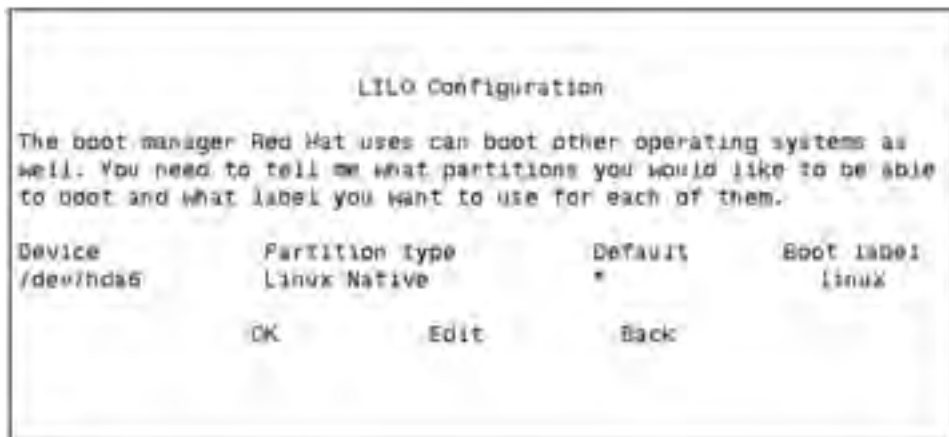
Because this system is going to run Linux, we are going to install the bootloader in the MBR (see Figure 3.58).

This code will construct the `/etc/lilo.conf` file with the following parameters:

```
[root@HostC /proc]# cat /etc/lilo.conf
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=linux
image=/boot/vmlinuz-2.4.2-2
    label=linux
    initrd=/boot/initrd-2.4.2-2.img
    read-only
root=/dev/sda6
```



**Figure 3.57** LILO configuration: bootloader.



**Figure 3.58** LILO configuration: boot manager.

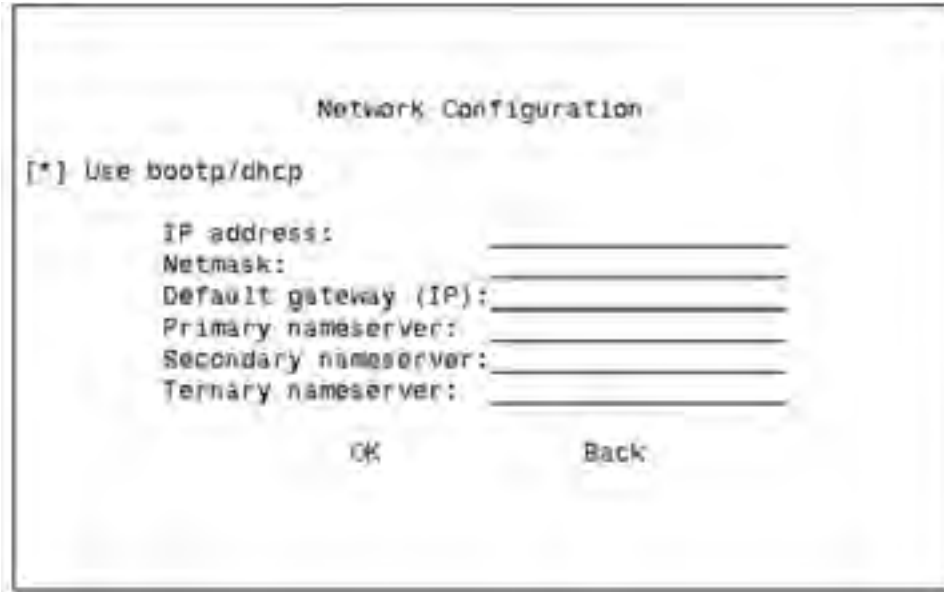
### Step 8. Network Configuration

The IP address information that we entered on this screen for Figure 3.59 is as follows:

#### DISABLE BOOTP/DHCP

IP address: 192.168.2.205  
 Netmask: 255.255.255.0  
 Default gateway: 192.168.1.1





**Figure 3.59** Network configuration.

Primary nameserver: 24.64.2.33  
Secondary nameserver: 24.64.2.34

This information was derived from our network administrator.

Now, we will set the Hostname of the system. The screen in Figure 3.60 will update both the `/etc/hosts` and `/etc/sysconfig/` network files with the hostname parameter.

### Step 9. Firewall Configuration

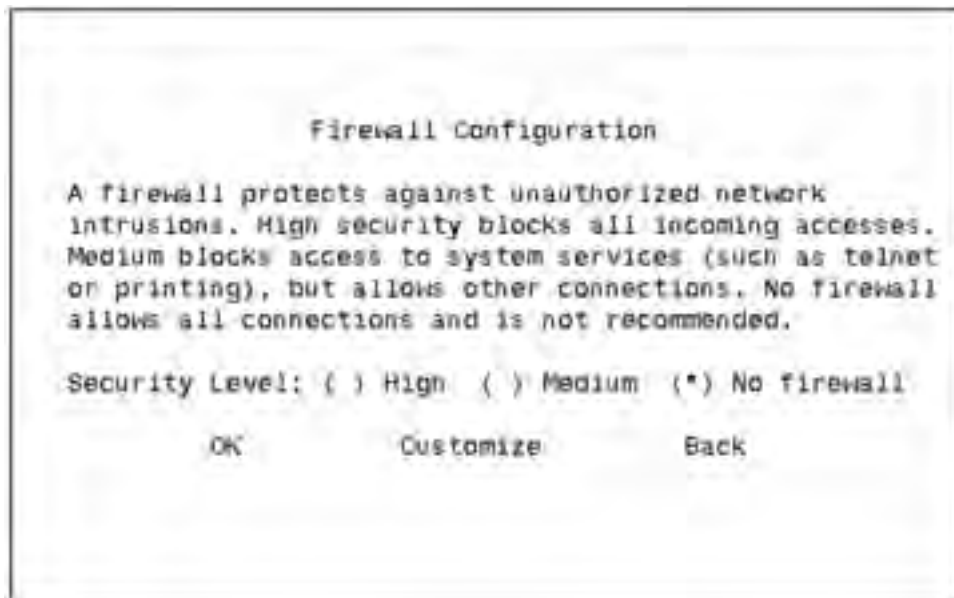
This particular system will be behind an existing firewall as well as inside a controlled data center. Therefore, for Figure 3.61 we will disable the firewall feature. If you are going to use this feature, make sure that you take the time to review the Customize option. Many daemons that you rely on will be effectively disabled as a result of enabling the firewall feature. For example, it disables the Telnet option.

### Step 10. Mouse Selection

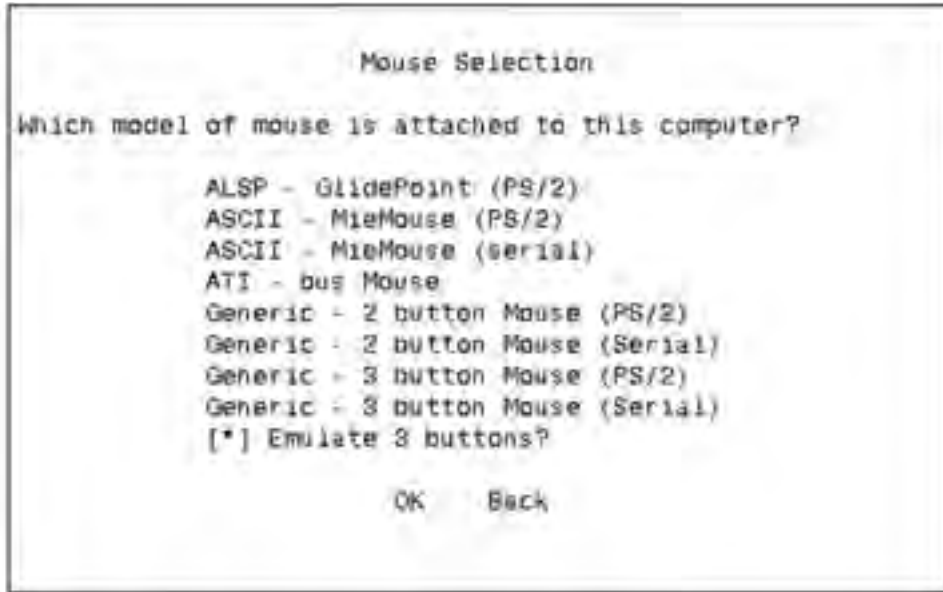
Here, for Figure 3.62, we chose Generic—2 button Mouse (PS/2) with the Emulate 3 buttons.



**Figure 3.60** Hostname configuration.



**Figure 3.61** Firewall configuration.



**Figure 3.62** Mouse selection.

### Step 11. Language Support

For Figure 3.63 we used English (USA).

### Step 12. Time Zone Configuration

Our time zone is Canada/Mountain, and the hardware clock set to GMT was disabled because we have the local time set in our system CMOS. Therefore, for Figure 3.64 we chose Canada/Mountain and GMT.

### Step 13. Root Password

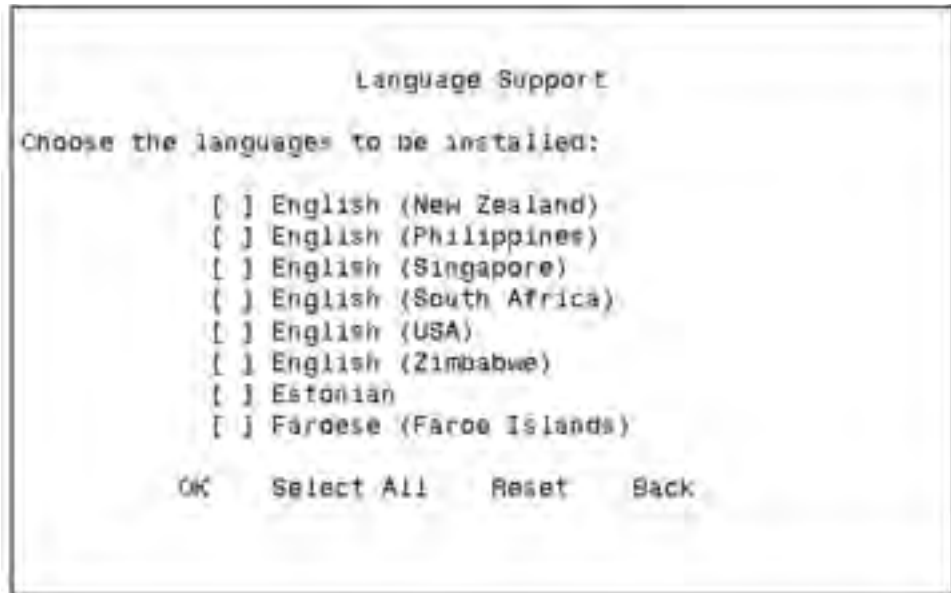
For Figure 3.65 use a good password, and make note of it.

### Step 14. Add Non Root Users

For Figure 3.66 we chose to create only one non-privileged user, knowing we could create users later.

### Step 15. Authentication Configuration

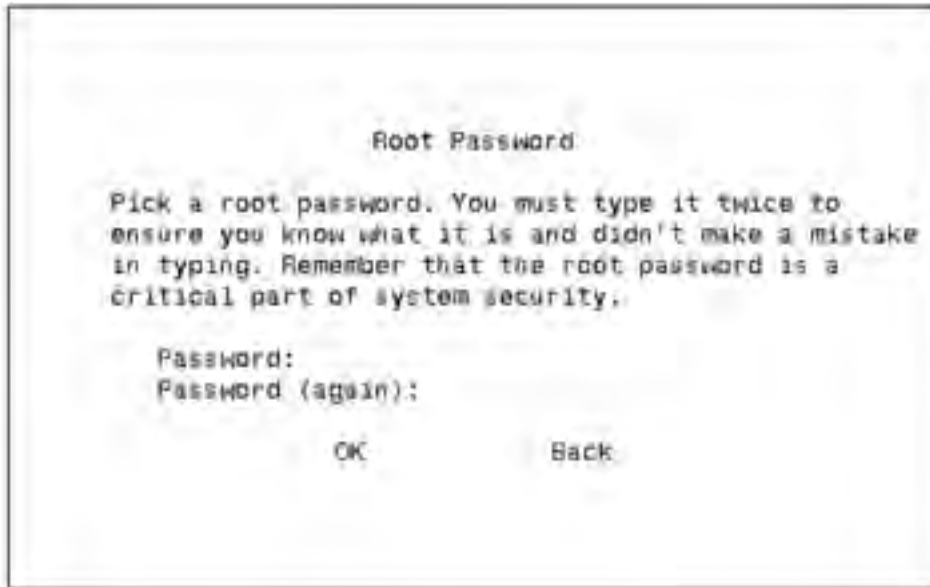
Accept and use the defaults here. It is our intention to use local authentication as opposed to a remote LDAP or Kerberos server environment. For Figure 3.67 we chose to use local authentication.



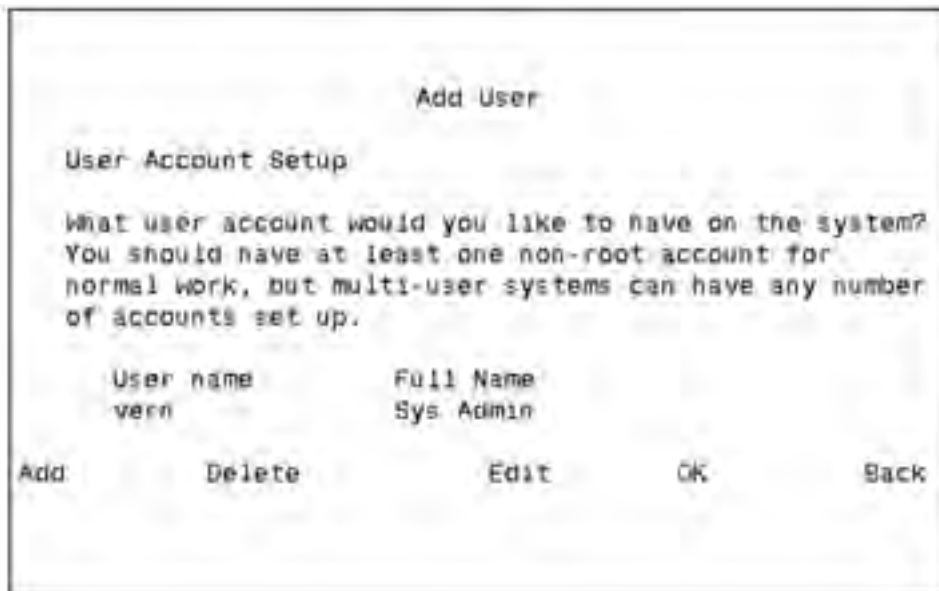
**Figure 3.63** Language support.



**Figure 3.64** Time zone configuration.



**Figure 3.65** Root password.



**Figure 3.66** Add user.



**Figure 3.67** Authentication configuration.

### Step 16. Package Group Selection

A specific set of packages was installed on this system (as listed) plus some additional individual selections that we know we need. Remember that you can install additional packages at any time after the installation. The reason we choose to install all of the User Interface/X Hardware Support for Figure 3.68 is because we do not know which one we are going to require. After our video is configured, we will remove those that we did not require:

- Main packages
- Printer
- X Window System
- KDE
- DOS/Windows connectivity
- Multimedia support
- Networked workstation
- NFS Server
- Web Server
- Development
- Kernel development



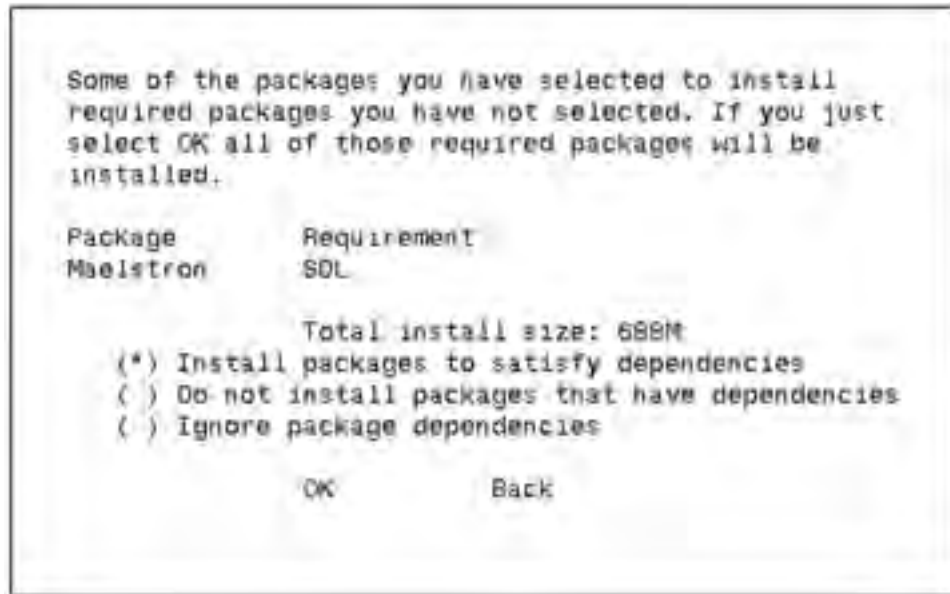
**Figure 3.68** Package group selection.

- Utilities
- Package group selection
- Applications/system
- gnome-linuxconf
- linuxconf
- User Interface/X
- XFree86-xf86cfg
- User Interface/X Hardware Support
- All

Figure 3.69 shows the package requirements. Definitely choose OK to proceed.

### Step 17. Video Configuration

The video configuration screen is shown in Figure 3.70. X probe has detected a Cirrus Logic chip inside the system. We recommend that you write down



**Figure 3.69** Package requirements.



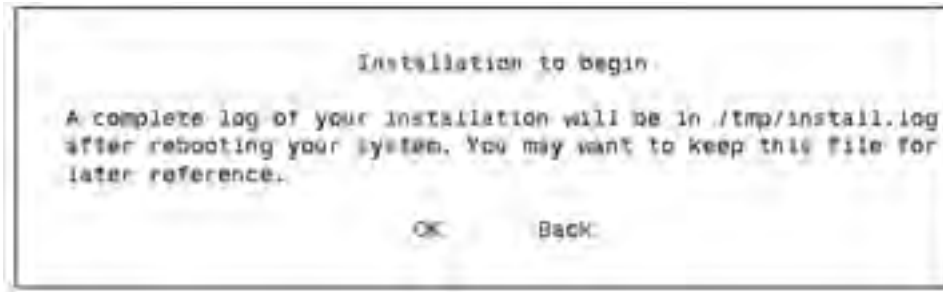
**Figure 3.70** X Probe results.

these details at this time in the event that the video configuration does not cooperate when you actually try to start X Windows.

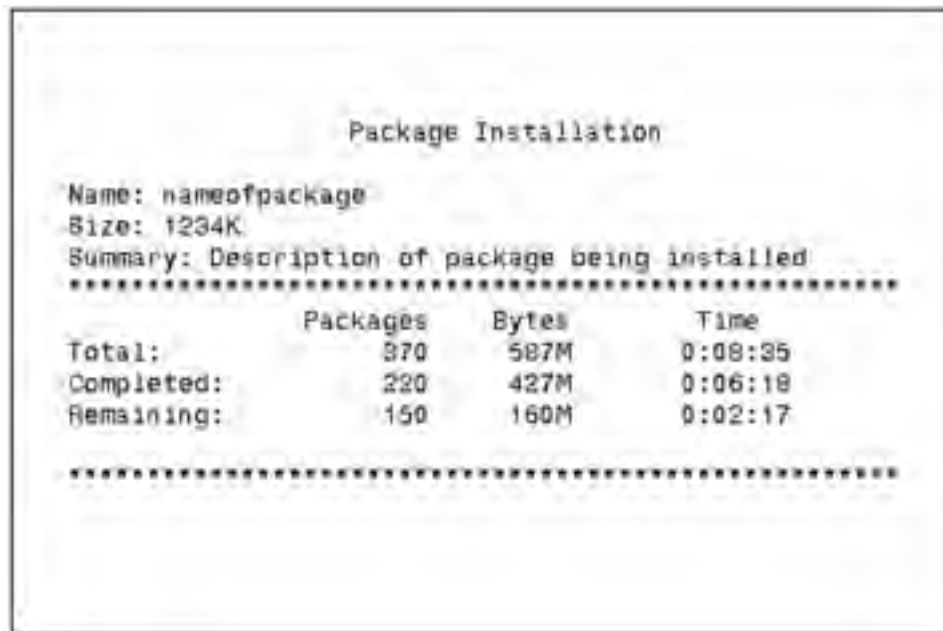
### Step 18. Installation

Choose OK to continue (see Figure 3.71). The package installation will proceed as shown in Figure 3.72.





**Figure 3.71** Install to begin.

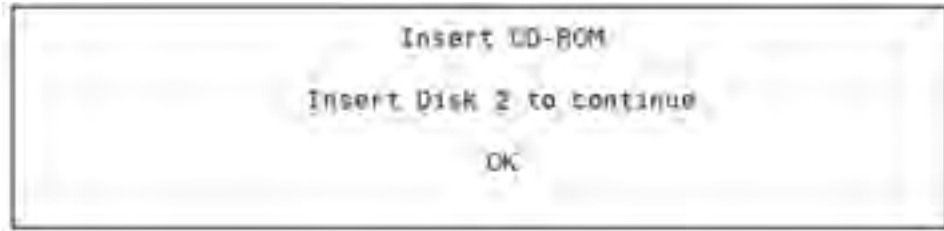


**Figure 3.72** Package installation.

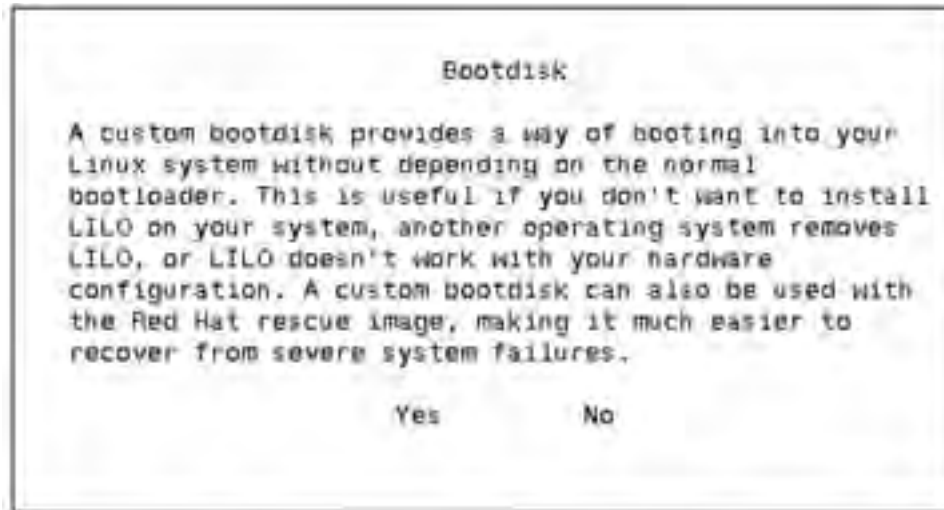
Insert the second CD and press OK. (see Figure 3.73). This brings up the bootdisk screen (see Figure 3.74).

### **Step 19. Create a Custom Bootdisk**

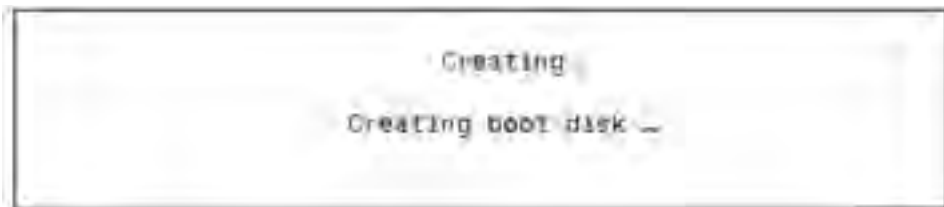
Create the bootdisk and label it appropriately. Figure 3.75 shows the command in action.



**Figure 3.73** Insert CD-ROM, disk 2.



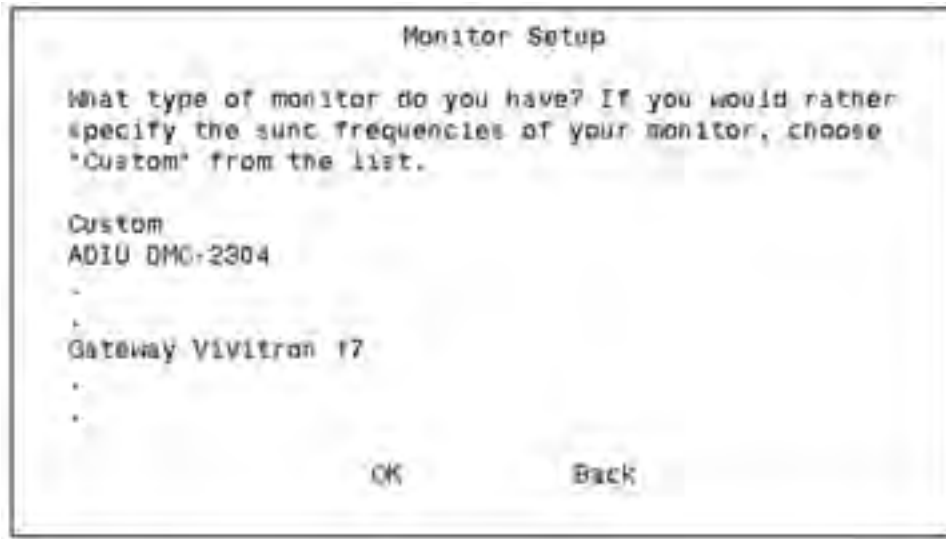
**Figure 3.74** Bootdisk-request.



**Figure 3.75** Bootdisk-creating.

## Step 20. Monitor Setup

This setup is one of the most interesting processes in the entire installation process (see Figure 3.76). Let's explain the behavior here. The monitor



**Figure 3.76** Monitor setup.

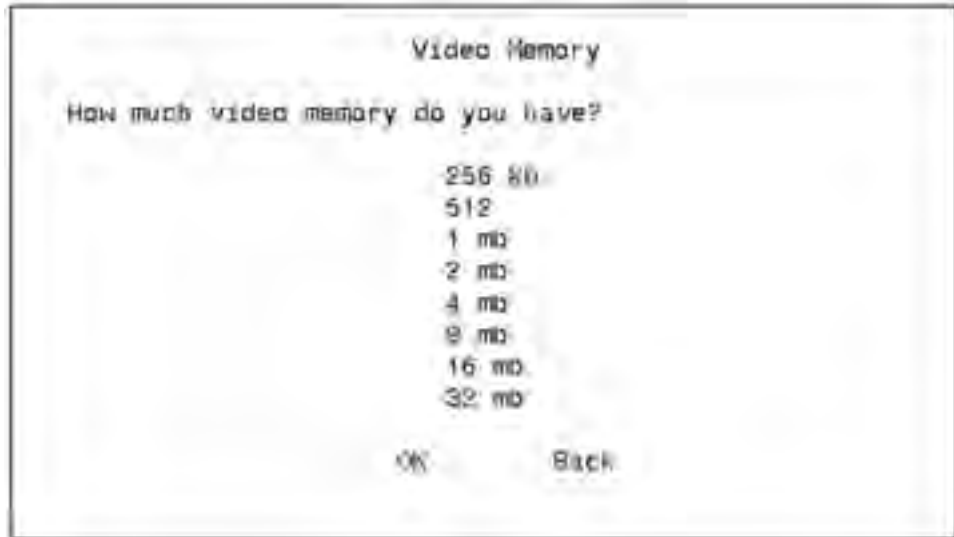
setup did not detect our monitor. If it did, it would have presented a screen stating that it had located a Dell Ultrascan 21TE. As we navigate the list, however, we see that our monitor type is indeed listed. We will choose the Dell Ultrascan 21TE option, of course, but based on experience we definitely have to revisit this system and set it up as a Custom monitor. Fortunately, we have the hardware manual and specifications for this particular monitor. Do not force the issue here, just move on. There are better tools to configure video and monitors that will be available when the system is running. We cover this in an alternate chapter.

### **Step 21. Back to Video Configuration**

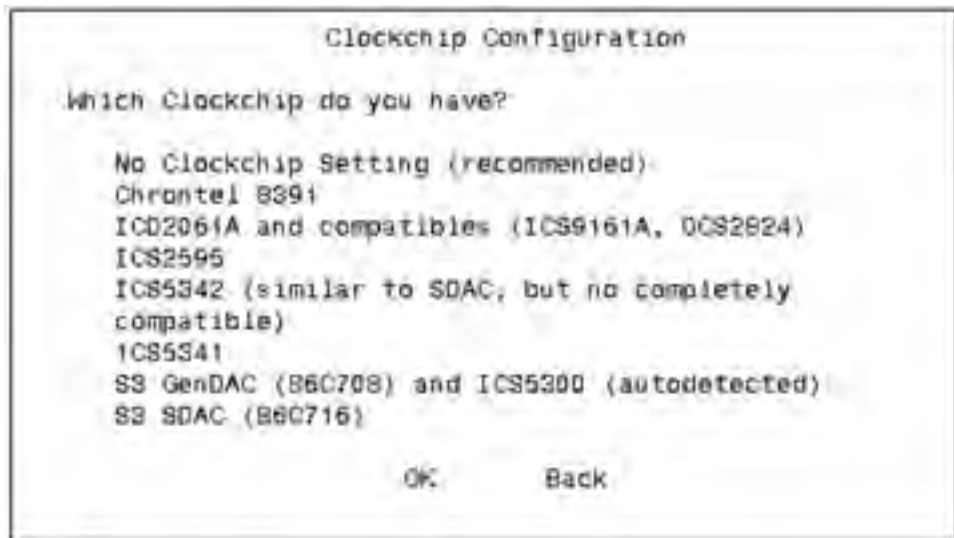
Figure 3.77 shows the video memory screen. Because the previous video detection failed, this procedure is not going to be of any value. Therefore, it does not matter what it sees or whether it is accurate or not. Just press OK and continue to the clockchip configuration (see Figure 3.78).

We offer the same comment as we did earlier—choose No Clockchip Setting (recommended) to move on to the video modes screen (see Figure 3.79).

Although we are going to configure the video setting later, you have to choose at least one here. We recommend that if you choose anything,

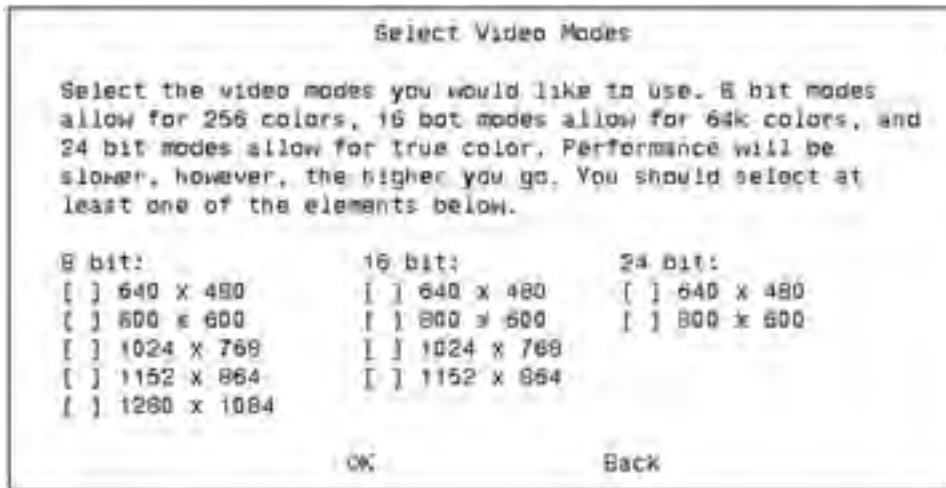


**Figure 3.77** Video memory.

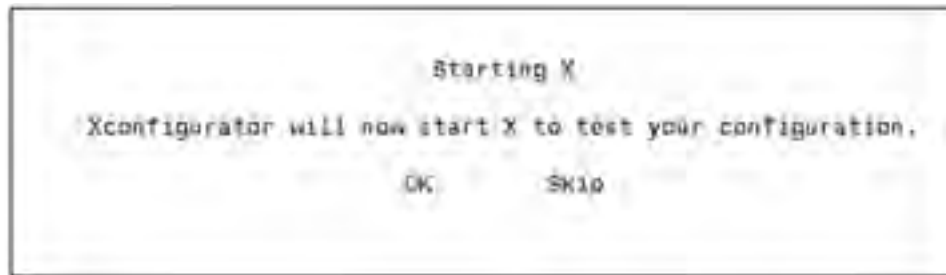


**Figure 3.78** Clockchip configuration.

choose at least these: 8 bit:  $800 \times 600$ ,  $1024 \times 768$ , 16 bit:  $800 \times 600$ , and  $1024 \times 768$ . These are representative of the lowest common denominator (what most monitors can handle).



**Figure 3.79** Select video modes.



**Figure 3.80** Starting X.

## Step 22. X Window Configuration

Figure 3.80 shows the window configuration screen. This action is going to fail, but do not worry. You can choose *OK* here just for fun at this point, but if your video is not going well, this procedure will definitely not work. If it works, then choose *Yes*. Only choose *Yes* if you do not see any irregularities. A typical irregularity would be a box or small images hanging around the mouse icon or black lines on the screen where they should not be. Respond *No* if there are any irregularities.

**WARNING** You will be presented with a screen asking you whether you would like the X window desktop to automatically start when the system starts. We highly recommend that you say *No* to prevent the desktop from hanging up

the system. You can always start the X window desktop at any time by entering `startx` at the command line after you log into the system.

### Step 23. Complete.

The final screen for completion is shown in Figure 3.81. Reboot, and you should get a login prompt.

## *Troubleshooting the SCSI Installation*

Problems with SCSI drives often are a result of not having the SCSI bus set up correctly. For example, it is best practice to have the SCSI controller set with a SCSI ID of 7 or the suggested manufacturer defaults. Double-check the termination rules for all the devices. In most configurations, the SCSI controller card needs to be terminated. The only time this situation is not the case is if the SCSI controller is in the middle. This situation would be the case if some of the SCSI drives are internal and some are external. Alternatively, the SCSI controller card is self-terminating. The only way you would know is to read the documentation that came with the card. It is up to you to make sure that each SCSI drive or device has a different SCSI ID configured on it. For example, knowing that the SCSI controller adapter is using SCSI ID 7, configure the SCSI hard disk to use SCSI ID 5 and the CD-ROM to use SCSI ID 3. SCSI IDs also infer order to the SCSI



**Figure 3.81** Complete.

protocol. We recommend that you read about the SCSI protocol before proceeding to work these kinds of drives. The SCSI device on the end, which will be either the hard drive or the CD-ROM in our example, needs to be terminated as well—not both, and it has to be the one on the end. So, make sure that the device in the middle of the SCSI chain is not (repeat, *not*) terminated. Think of it as such: SCSI device A with termination, SCSI device B no termination, SCSI device C no termination, and SCSI device D with termination. The same rule applies to SCSI as applies to IDE. Do not mix different SCSI technologies. For example, do not connect a SCSI fast and wide device to an older SCSI I controller. Again, it might work, but ask yourself if you are being smart or lucky.

### ***Confirm and Test the System***

These are just very basic checks that we recommend before you invest many hours configuring applications and creating users. The entire installation process takes only an hour or two and if anything needs to be addressed, now is the time to do it.

### **Networking**

Check the networking first. Most software on the system relies on a proper TCP/IP configuration to function properly. Perform a simple ping test. Ping your internal address. If you cannot see your own network card, then you have to revisit your network card configuration. Start by running the `netcfg` utility and see whether the card is active. Sometimes this problem is all that it is, and it only takes a second to check before you change any parameters.

If your X Windows is not working but it will present an equivalent text menu version, another file that you can check is `/etc/sysconfig/hwinfo`. If the device fails to appear in the `/etc/sysconfig/hwinfo` file, then you have a hardware conflict or dead card or perhaps a BIOS problem.

### **Video Configuration**

You can run a utility called `/usr/bin/X11/SuperProbe` to find out what kind of video chipset you have. Also check the `/var/log/Xfree86.x.log`, where *x* is the current occurrence of the log. If this utility cannot detect the chipset, then head directly to the Red Hat support site and check the hardware compatibility list again. A troublesome video configuration can take quite some time to resolve. We are going to refer you to our Chapter 5, “Configuring the Video.”

### Review the Logs

You should review the contents of the following logs for irregularities:

```
/tmp/install.log  
/var/log/boot.log  
/var/log/dmesg  
/var/log/messages
```

### Document the System

This task is easy. Most of the information that you need is currently sitting in the */proc* and */etc/sysconfig* directories under various directories and files labeled by their device names. It is a good idea to print these files and put them in a binder in a safe place. We encourage that you read these files. At a minimum, you should tar up these directories and keep them somewhere in order to access them. This material should also be part of your backup and recovery plan. If you have to rebuild the system or a similar system, these can be a very valuable resource.

## Reconfiguring the System

---

There is a script called */usr/sbin/sys-unconfig* that can be used to reconfigure the system. This script creates a zero-byte */.unconfigured* file and halts the system. You will have to restart the system. When the system is restarted, the */etc/rc.d/rc.sysinit* script will systematically guide you through reconfiguring the main parameters of the system. The script will run *passwd* (to change the root password), *netconfig*, *timeconfig*, *authconfig*, and *ntsysv*. This feature is very handy if you are moving a server from one location to another.



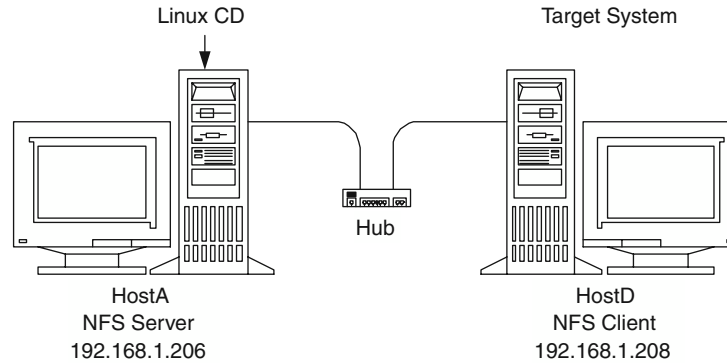




## Installing Linux by Using NFS

In Chapter 3, we covered installing Linux on SCSI and IDE systems. For both installations, we used a local CD-ROM as the source for the operating system files. In this chapter, we install Linux remotely by using a network connection that uses NFS. We will set up an NFS server as a source system and an NFS client as a target system. This situation is ideal for labs or facilities that need to install many Linux systems over the network routinely. This setup also enables you to customize the RPMs. Once the RPMs are copied to the hard disk, they can be updated and supplemented to create a standard build. A standard build is defined as a complete set of RPMs that are known to work together in a very standardized and stable environment. Figure 4.1 shows the configuration that we are describing in the following instructions.

All of the necessary RPMs and binaries will be copied to the NFS server, and the installation procedure will take place on the NFS client. When the NFS client is installed with Linux, it will be a standalone Linux system and will no longer be an NFS client. An additional topic is added to the end of this unit that explains how to perform a casual NFS mount that does Unix network file sharing.



**Figure 4.1** NFS server and NFS client.

There are two major tasks in this type of installation. First, prepare the NFS server, and second, prepare the NFS client. The NFS client will end up being a new Linux host after the installation completes. Of course, in our example, the NFS server is another Red Hat Linux system.

## Quick Review of Configuration

---

Sun Microsystems developed NFS in the 1980s and courteously shared it with the industry so that NFS became a standard way for sharing disk resources across a network. Its primary strength is its interoperability. You can find NFS support in every UNIX-related operating system, and for just a few dollars (comparatively), you can get NFS support for other operating systems, as well. Microsoft has recently announced support for NFS on Windows 2000 server operating systems. Its second strength is its capability to make file systems on remote systems appear to be local on client systems on the network. NFS is generic, and once you understand the basics, you can apply them to other operating systems as well.

There are a couple of network requirements, however, and here we are going to rely on the classic non-UNIX definitions of server and client to explain. Specifically, a server is a system that responds to commands from clients, providing services such as data management, network administration, and security. A client is a standalone system that accesses shared network resources provided by a server. Be careful, because these are not Linux/UNIX definitions.

Referring to Figure 4.1, the network requirements are as follows. HostA and HostB must be capable of communicating via the TCP/IP protocol

stack. HostA, which is the system with the exported binary files already, installed either on its hard disk. Red Hat version 6.X had all the necessary RPMs on a single CD, which made it possible just to mount the CD and export it. With Red Hat version 7.X, however, the RPMs are spread over two CDs. This situation is problematic for NFS because the installation process requires the second CD, and there is no elegant way to unmount and remount the CDs without breaking the NFS connection. As a result, we recommend that you copy the RPMs from the two respective CDs into a single directory on HostA and export that directory. We will document this situation for a few reasons. First, installing from a hard disk is faster and more efficient. Second, once the required directories are copied from the CD to the disk, you can update them. This process will reduce the time spent afterward upgrading RPMs because the correct ones were used at the beginning. Third, you can install as many clients as you want simultaneously as long as you have sufficient bandwidth to do so.

## NFS Configuration Details

---

NFS, as a protocol, has a concept of security and ownership. For the purposes of this exercise, we will not configure security on the NFS server. We will export the file system in such a way that any NFS client on the network can access it. If you are going to share file systems routinely with other users on other systems, you have to ensure to the best of your ability that they are trustworthy and are set up using the basic security features of NFS. Otherwise, you might want to consider an alternative syntax to that documented here to provide better security. For more information on NFS, syntax please refer to the man pages.

## Preparing the NFS Server

Set up an NFS server by following these steps. Note that these instructions assume that you are on HostA, which is going to be your NFS server.

### **Step 1. Insert the Linux CD-ROM in HostA (NFS Server)**

Put the Linux operating system CD-ROM into the CD-ROM drive of your network's designated NFS server. As discussed previously, this server could be any system on which NFS is supported regardless of the installed operating system—Linux, another UNIX flavor, DOS, OS/2, and so on—as

long as the system can function as an NFS server. The system in our example is a server with Linux installed on it.

### **Step 2. Log in as a Regular User and su to root**

Log in to the Linux NFS server (HostA) as a regular user and su to root, because you will have to verify that the NFS daemons are installed and running. You also have to modify one of the *tcp* files in the */etc* directory.

```
Red Hat Linux release 7.1 (Seawolf)
Kernel 2.4.2-2 on an i686
login: linda
Password:
Last login: Fri Jun 22 14:12:55 from 192.168.1.101
[linda@HostA linda]$ su - root
Password:
[root@HostA /root]#
```

### **Step 3. Verify that the NFS Package Is Installed**

```
[root@HostA /root]# rpm -qa | grep -i nfs
nfs-utils-0.3.1-5
```

If NFS is not installed, install the package now.

```
[root@HostA /root]# mount /dev/cdrom
[root@HostA /root]# cd /mnt/cdrom/RedHat/RPMS
[root@HostA RPMS]# rpm -install nfs-uils-0.3.1-5.i386.rpm
```

### **Step 4. Verify that the NFS-server daemons Are Running**

```
[root@HostA /root]# ps aux | grep -i nfs
root      855  0.0  0.0    0   0 ?        SW   08:34   0:00 [nfsd]
root      856  0.0  0.0    0   0 ?        SW   08:34   0:00 [nfsd]
root      859  0.0  0.0    0   0 ?        SW   08:34   0:00 [nfsd]
root      860  0.0  0.0    0   0 ?        SW   08:34   0:00 [nfsd]
root      861  0.0  0.0    0   0 ?        SW   08:34   0:00 [nfsd]
root      862  0.0  0.0    0   0 ?        SW   08:34   0:00 [nfsd]
root      863  0.0  0.0    0   0 ?        SW   08:34   0:00 [nfsd]
root      864  0.0  0.0    0   0 ?        SW   08:34   0:00 [nfsd]
root     1413  0.0  0.0  1620  592 pts/0    S    14:16   0:00 grep -i nfs
[root@HostA /root]#
```

If they are not started, start them now. The easiest way to start daemons is to locate the correct start file. Navigate to the run-time directory, locate the start script, and start it by using the *S60nfs* script. Once you become

more familiar with the scripts, you can just run it by using the complete path. For now, we will use this technique:

```
[root@HostA /root]# cd /etc/rc3.d
[root@HostA rc3.d]# ls
K01kdcrotate  K50snmpd      S10network    S40atd         S85gpm
K03rhnsd      K50tux        S12syslog     S55sshd        S90crond
K15httpd      K65identd     S13portmap    S56rawdevices  S90xfs
K20rstatd     K73ypbind     S14nfslock    S56xinetd      S95anacron
K20rusersd    K74nscd       S17keytable   S60lpd         S99local
K20rwalld     K74ypserv     S20random     S60nfs
K20rwhod      S05kudzu      S25netfs      S80isdn
K34yppasswdd  S08ipchains   S26apmd       S80pppoe
K45arpwatch   S08iptables  S28autofs     S80sendmail

[root@HostA rc3.d]# ./S60nfs start
Starting NFS services:                                [ OK ]
Starting NFS quotas:                                  [ OK ]
Starting NFS mountd:                                  [ OK ]
Starting NFS daemon:                                  [ OK ]
[root@HostA rc3.d]#
```

### Step 5. Record the IP Address Details from the NFS Server

You will need this information later to provide the target installation system so that it can locate the exported directory. Use the following `ifconfig` command syntax:

```
[root@HostA /etc]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:BA:DB:A1:9D
          inet addr:192.168.1.206  Bcast:192.168.1.255
          Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10439 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3955 errors:0 dropped:0 overruns:0 carrier:0
          collisions:8 txqueuelen:100
          Interrupt:11 Base address:0xd800
[root@HostA /etc]#
```

### Step 6. Copy the Files to the Hard Disk

Check to make sure that you have at least 1.2GB of space for the source files. We have created a directory in the `/home` mount point called `/home/inst_files`, and we copied the RedHat directories from both CD 1 and CD 2 into `/home/inst_files`. The following listing shows how much disk space was required. You will have to copy the files from both the CDs. The files that

you need to copy from the CDs are in the RedHat directory. If your drive will not open, you might have to exit the directory and unmount the CD-ROM and remount the new CD.

```
[root@HostA /root]# mount /dev/cdrom
[root@HostA /root]# cp -R /mnt/cdrom/RedHat /home/inst_files/
[root@HostA /root]# cd
[root@HostA /root]# umount /dev/cdrom
[root@HostA /root]# mount /dev/cdrom
[root@HostA /root]# cp -R /mnt/cdrom/RedHat /home/inst_files/
[root@HostA /etc]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/hda8	251M	52M	186M	22%	/
/dev/hda1	53M	3.5M	46M	7%	/boot
/dev/hda6	1.6G	1.1G	545M	66%	/home
/dev/hda7	1.6G	804M	786M	51%	/usr
/dev/hda9	251M	19M	219M	8%	/var

```
[root@HostA /etc]#
```

### Step 7. Export the Directory by Using NFS; No Security

Having no security here is a compromise to ensure that those who are not familiar with NFS will have a successful export. No security here is denoted by the \* symbol, which implies that any host and not just HostB can mount this exported directory. This situation translates into giving access to the world. If security were enabled, the client that will be the Linux installation target (known as HostB) would have no way of authenticating itself to make the initial connection to the NFS server. In other words, any kind of security would stop you dead in your tracks at the beginning of the installation. To tell the NFS server to export the directory called */home/inst\_files*, start by adding the following entry to the */etc/exports* file. Then, use the *exportfs* command to actually export the directory.

```
[root@HostA /root]# cat /etc/exports
# This is the exports file for HostA
#
/mnt/cdrom      *(ro)
/home/inst_files  *(ro)
[root@HostA /root]#
```

The entries in this file will export the CD-ROM mounted on */mnt/cdrom* and the directory */home/inst\_files* as read-only with no other parameters.

```
[root@HostA /root]# exportfs
[root@HostA /root]# exportfs -a
```

```
[root@HostA /root]# exportfs
/home/inst_files    <world>
/mnt/cdrom          <world>
[root@HostA /root]#
```

You will receive some warnings that you can ignore. Your export was successful if you can see the `/home/inst_files <world>` response. The world entry signifies that anyone with an NFS client can mount this exported directory remotely. The `exportfs` command, when used with no arguments, just reports the status of the exports list. The `exportfs -a` command processes and exports all the directories in the `/etc/exports` file. Use `exportfs -ua` to unexport all entries regardless of what is in the `/etc/exports` file. When you are finished, you might want to remove the entries from the `/etc/exports` file.

To review, HostA is now running the NFS daemons and is currently exporting a directory called `/etc/exports`. This configuration is only 50 percent complete until the NFS client is built. Instructions for the NFS client are next.

## Preparing the NFS Client

In this case, the NFS client is for purposes of initiating the installation and providing a protocol path for the files to be copied from the NFS server (HostA) to the NFS client (HostB). After the installation is complete (it takes a few minutes), the NFS client will be rebooted (at that point, it will be a Linux host).

### Step 1. Create NFS boot install Disks

The process for creating this boot disk is identical to the process of creating a regular boot disk. Use the `bootnet.img` file rather than the `boot.img` file. Change to the `/mnt/cdrom/images` directory and list the contents of that directory to confirm that the `bootnet.img` file is present:

```
[root@HostA cdrom]# cd /mnt/cdrom/images
[root@HostA images]# ls -l
total 5333
-r--r--r--      1 root    root          485 Aug 30  2000 TRANS.TBL
-rw-r--r--      2 root    root    1474560 Aug 30  2000 boot.img
-rw-r--r--      2 root    root    1474560 Aug 30  2000 bootnet.img
drwxr-xr-x      2 root    root        2048 Aug 30  2000 de
-rw-r--r--      2 root    root    860160 Aug 30  2000 drivers.img
drwxr-xr-x      2 root    root        2048 Aug 30  2000 es
drwxr-xr-x      2 root    root        2048 Aug 30  2000 fr
drwxr-xr-x      2 root    root        2048 Aug 30  2000 it
```



-rw-r--r--	2	root	root	147456	Aug 30	2000	paride.img
-rw-r--r--	2	root	root	1474560	Aug 30	2000	pcmcia.img
drwxr-xr-x	2	root	root	2048	Aug 30	2000	pxeboot
drwxr-xr-x	2	root	root	2048	Aug 30	2000	sv

Run the `dd` utility to transfer the *bootnet.img* file to the floppy diskette:

```
[root@HostA images]# dd if=bootnet.img of=/dev/fd0
2880+0 records in
2880+0 records out
```

The result will be a bootable floppy with the following files on it that you can use to boot the system and to initiate the NFS installation process.

### **Step 2. Boot the Target System**

Go to the target installation system, HostB, which will be the NFS client. Insert the Linux boot diskette and proceed to the Installation Method screen. Note that this screen might vary slightly (that is, it will present a slightly different list of installation media selections) depending on whether you have booted with the Red Hat 5.2 or 6.x versions. The NFS Image selection appears regardless of the version you are using.

### **Step 3. Choose the NFS Image Option**

In the Installation Method screen, select the NFS Image option.

### **Step 4. Provide Networking Information**

You will be prompted next for networking information for the target system. This information is the IP address for the NFS client system known as HostB—that is, the IP address for the system from which you are initiating the installation. Do *not* enter the IP address of the NFS server yet; this step will come later. What it is trying to do is enable the network card inside the target system so that it can communicate by using TCP/IP, and it needs to have this information now.

**IP address:** 192.168.1.208

**Netmask:** 255.255.255.0

**Default gateway (IP):** 192.168.1.1

**Primary nameserver:** 24.64.2.33

We used the suggested defaults for the default gateway and the primary nameserver because these two systems are on the same segment, and what you use for their addresses will not make any difference at this point. *The only*

two entries that you need are the IP address and the netmask; the default gateway (IP) and primary nameserver can be reconfigured when the host reboots.

After providing the preceding information, you will be asked for the intended hostname for the NFS client.

**Domain name:**

**Host name:** `HostB`

**Secondary nameserver (IP):**

**Tertiary nameserver (IP):**

We enter `HostB` here because our network is not configured for local name services; we are using flat `/etc/hosts` files for all our servers. At this point, you should be able to ping between the two systems. At the NFS client system, you can use the Alt-F1 and Alt-F2 keys to toggle back and forth between the screens to get a command line that will enable you to issue a ping command.

If you have not already guessed, this *bootnet.img* file has a limited number of supported network drivers with which you can work. It is mandatory at this point that you use only formally supported and listed network cards for the NFS client.

### **Step 5. Provide Information about NFS Server**

Next, you will be asked for information on the NFS server that you set up. This information is the IP address for the NFS server that you recorded in the previous set of steps. What it will do here is construct a `mount` command using the IP address you give it here and the name of the exported file system. In other words, it will mount `/home/inst_files` on system 192.168.1.206, and at this point the `/home/inst_files` directory appears to be a local entity over NFS between the two systems.

**NFS server name:** `192.168.1.206`

**Red Hat directory:** `/home/inst_files`

At this point, the target installation system will use this information to construct a mount to the NFS server. If things do not go well, use the Alt-F1, Alt-F2, and Alt-F3 keys to toggle back and forth between screens and a command line that will enable you to issue a ping command.

### **Step 6. Proceed with Otherwise “Normal” Installation**

The rest of the installation will continue normally.

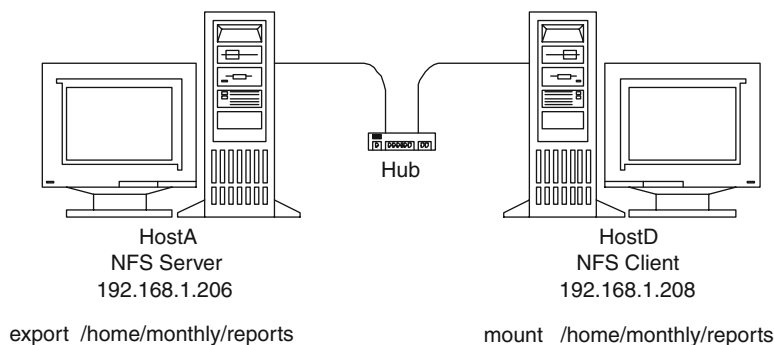
## Typical NFS File Sharing

We set up the previous scenario to facilitate a remote installation. NFS is more commonly used as a way to share Unix-type file systems over a network. We have prepared a companion scenario whereby we will export a directory called `/home/monthly/reports` from HostA and have HostB mount it. Once HostB has mounted the HostA NFS directory over the network, it is available for use on HostA. Often, these mounted directories are used by both application software and users (see Figure 4.2).

A simple scenario has been created to open discussion. Three users named Sancho Panza (username `panzasan`); Don Quixote (username `quixoted`); and Lady Dulcinea (username `dulcinea`) have been created on HostA as examples. Fernando (username `fernando`) is the boss and has been created on HostB. All three are famous characters from the 1964 novel *Don Quixote* by Miguel de Cervantes, London: MacMillan & Co, Ltd.

Review the following information and make note of two things in particular. First, note their assigned uids and gids. Their uids and gids are in the 500 range, which denotes a non-privileged user. Second, note that they are on HostA. The uids and gids play an important role in setting up NFS. We will return to how they are used later.

```
[root@HostA RPMS]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
```



**Figure 4.2** NFS file sharing.

```

halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/var/ftp:
nobody:x:99:99:Nobody:/:
nscd:x:28:28:NSCD Daemon:/bin/false
mailnull:x:47:47:/var/spool/mqueue:/dev/null
ident:x:98:98:pident user:/bin/false
rpc:x:32:32:Portmapper RPC user:/bin/false
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/bin/false
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
apache:x:48:48:Apache:/var/www:/bin/false
linda:x:500:500:/home/linda:/bin/bash
panzasan:x:501:501:/home/panzasan:/bin/bash
quixoted:x:502:502:/home/quixoted:/bin/bash
dulcinea:x:503:503:/home/dulcinea:/bin/bash
keeler:x:505:505:/home/keeler:/bin/bash
teamxx:x:506:506:/home/teamxx:/bin/bash
teamyx:x:507:507:/home/teamy:/bin/bash
[root@HostA RPMS]#

```

The directory that will be shared contains monthly report information. Each report was created by each user in their home directory, then copied over to the `/home/monthly/reports` directory. The base permissions on the directory are listed. The permissions on the `/home/monthly/reports` directory will enable anyone who is part of the reports group to copy a file to the directory. All of these users are members of the reports group. It is also necessary to note that each file in this directory is owned by each user, respectively.

```

[root@HostA /root]# ls -l /home/monthly
total 4
drwxrwx--- 2 root      reports    4096 Jun 22 18:44 reports
[root@HostA /root]# ls -l /home/monthly/reports
total 12
-rw-rw-r-- 1 dulcinea dulcinea    113 Jun 22 18:41 dulcinea_report
-rw-rw-r-- 1 panzasan panzasan    100 Jun 22 18:44 panzansan_report
-rw-rw-r-- 1 quixoted quixoted     88 Jun 22 18:43 quixoted_report
[root@HostA /root]#

```

This code represents the directory and the information to be exported from HostA. Unlike the previous example, we are only going to let HostB mount the directory. The `/etc/exports` file refers to HostB by name; therefore,

it is necessary to update the */etc/hosts* file so that NFS can resolve the host alias HostB to the IP address 192.168.1.208. An alternative would have been to make sure that HostB could be resolved via DNS.

```
[root@HostA /root]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
192.168.1.208      HostB
192.168.1.206      HostA
192.168.1.222      HostC

[root@HostA /root]# cat /etc/exports
# This is the exports file for HostA

/mnt/cdrom *(ro)
/home/inst_files *(ro)
/home/monthly/reports HostB(rw)
[root@HostA /root]#
```

The entry in the */etc/exports* file states that HostB has read-write access to */home/monthly/reports*. The NFS assumption here is important. The assumption is that if there is no host defined or a \* symbol, then all hosts have access. Nevertheless, once a single host is defined, then only that host can have access and all others are denied. Therefore, the top two export statements will allow any host, but the bottom entry is restricting access to a system called HostB (which is 192.168.1.208). NFS offers only moderate security. For example, it is possible that someone could wait for you to take your system down or take your system down. Then, they can spoof your HostA system by bringing up another system with the IP address 192.168.1.208 with a hostname of HostB and posing as your other system, thus violating the trust relationship. For this reason, system administrators restrict the use of NFS. NFS systems are generally inside a firewall or a more secure environment that can be controlled. There are only two permission levels here: read-only and read-write. The reason we have set up this user scenario is to demonstrate how read-write behaves.

The following examples show how to stop and start the NFS daemons and refresh the system to recognize the changes made to the */etc/exports* file:

```
[root@HostA /root]# cd /etc/rc3.d
[root@HostA rc3.d]# ./S60nfs stop
Shutting down NFS mountd:          [ OK ]
Shutting down NFS daemon:          [ OK ]
Shutting down NFS services:        [ OK ]
```

```

Shutting down NFS quotas: [ OK ]
[root@HostA rc3.d]# ./S60nfs start
Starting NFS services: [ OK ]
Starting NFS quotas: [ OK ]
Starting NFS mountd: [ OK ]
Starting NFS daemon: [ OK ]
[root@HostA rc3.d]#
[root@HostA /root]# exportfs -ua
[root@HostA /root]# exportfs -a
[root@HostA /root]# exportfs
/home/monthly/reports HostB
/home/inst_files <world>
/mnt/cdrom <world>
[root@HostA /root]#

```

HostA can now mount HostB's */home/monthly/reports* directory. The next examples are all taken from HostB, which is the NFS client. First, confirm that the prerequisites required to be an NFS client are in place:

```

[alan@HostB alan]$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 localhost.localdomain localhost
192.168.1.206 HostA
192.168.1.208 HostB
[alan@HostB alan]$ rpm -qa | grep -i nfs
nfs-utils-0.3.1-5
[alan@HostB alan]$ ping 192.168.1.206
PING 192.168.1.206 (192.168.1.206) from 192.168.1.208 : 56(84) bytes of
data.
Warning: time of day goes back, taking countermeasures.
64 bytes from 192.168.1.206: icmp_seq=0 ttl=255 time=672 usec
64 bytes from 192.168.1.206: icmp_seq=1 ttl=255 time=636 usec
64 bytes from 192.168.1.206: icmp_seq=2 ttl=255 time=614 usec
64 bytes from 192.168.1.206: icmp_seq=3 ttl=255 time=655 usec
64 bytes from 192.168.1.206: icmp_seq=4 ttl=255 time=680 usec

--- 192.168.1.206 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.614/0.651/0.680/0.033 ms
[alan@HostB alan]$

```

Before you can mount an NFS file system or directory on a local system, you have to create a mount point. The *mount point* should be an empty directory, because when you mount the *HostA:/home/monthly/reports* exported directory it will cover any directory used as a mount point. In other words, if you chose to mount *HostA:/home/monthly/reports* on */home/alan*, then user alan

would no longer see the files in his home directory. Some system administrators create a series of directories called */mnt1* and */mnt2* (and so on). Other system administrators use other means. As the next step, we are going to create an empty directory called */home/final*. Then, we are going to mount *HostA:/home/monthly/reports* on */home/final*. Therefore, the remotely mounted directly will become */home/final/monthly/reports* on HostB.

```
[root@HostB /root]# mkdir /home/final
```

There are a couple of ways to mount a remote file system over NFS. You can create an entry such as the following in the */etc/fstab* file and then issue a mount command. Alternatively, you can just type the mount command complete with entries at the command line. This knowledge is useful for temporary mounts, because there is no entry in the */etc/fstab*. Command syntax for the latter looks like the following:

```
[root@hostb /root]# mount -t nfs HostA:/home/monthly/reports /home/final
```

If this mount will be used repeatedly, it would be appropriate to make an entry in the */etc/fstab* file. This way, the file system can be mounted automatically by the system every time it boots.

```
[root@hostb /root]# cat /etc/fstab
LABEL=/ / ext2 defaults 1 1
LABEL=/boot /boot ext2 defaults 1 2
/dev/fd0 /mnt/floppy auto noauto,owner 0 0
none /proc proc defaults 0 0
none /dev/pts devpts gid=5,mode=620 0 0
/dev/hda6 swap swap defaults 0 0
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,kudzu,ro 0 0
HostA:/home/monthly/reports /home/final nfs exec,dev,suid,rw 1 1
[root@hostb /root]# mount /home/final
[root@hostb /root]# mount
/dev/hda5 on / type ext2 (rw)
none on /proc type proc (rw)
/dev/hda1 on /boot type ext2 (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
automount(pid627) on /misc type
autofs(rw,fd=5,pgrp=627,minproto=2,maxproto=3)
HostA:/home/monthly/reports on /home/final type nfs
rw,addr=192.168.1.206)
[root@hostb /root]#
```

When you get used to the syntax, NFS mounts are easy and have many uses. System administrators usually keep all of the software and utilities that they use daily on a single system. When they need access to the files, they will quickly set up an NFS export and mount. It is typical for applications to utilize NFS file system mounts. For example, several Web servers in a Web server farm can all have access to the same Web root for content instead of copying the content to each docroot. Database servers can be configured to export the directories that have the database to a series of database clients to balance the load.

One thing you have to be aware of is the nature of the communication between an NFS server and an NFS client. The NFS client is dependent on the NFS server, and if the NFS server shuts down or becomes unavailable, the NFS client daemons will relentlessly retry the connection. They retry the connection to the point where the NFS client's utilization will be so high that the system is unusable. You can remedy the situation somewhat by adjusting some timing parameters, but the real solution is to ensure that the NFS server remains available. Because of these dependencies and timing issues, NFS mounts should not be performed over unstable network mediums or over routers that could slow down the connection. You have to start maintaining records of which systems are exporters and which systems are mounters so that you can make sure that all mounts are taken down before either system is shut down. For example, before you shut down an NFS client, you should unmount the NFS file system first. Before you shut down an NFS server, you have to unmount all of the NFS file systems at the NFS clients.

There are permission issues involved. At this point, if you are using our example to create an NFS mount, you will realize that any user, including the root user, cannot access the mounted directory. This behavior is normal for NFS, and there are ways to deal with it. One way would be to revisit the NFS server and re-export the directory, allowing root access. By default, it is assumed that the root user on the NFS client system shall not have access to the file system unless you specify so. When you think about it more, it makes perfect sense. Why should we assume that the root user on HostB could manipulate permissions and structure of the NFS file system that really is owned by HostA? This situation is normal unless you specified during the export that root on HostB can have access.

```
[alan@hostb alan]$ su - root
Password:
[root@hostb /root]# cd /home/final
```



```
bash: cd: /home/final: Permission denied
[root@hostb /root]#
```

This situation also holds true for users, unless the user owns the file. Owning the file in Unix means that uid and or gid that you are logged in with matches the uid and or gid of the file. In the following example, we are logged into HostB as user fernando. The `id` command shows that fernando's uid is 501 and this gid is 501, and he is a member of a group called reports with a gid of 508. This information is very significant and poses privacy and security problems.

```
[fernando@hostb fernando]$ id
uid=501(fernando) gid=501(fernando) groups=501(fernando),508(reports)
[fernando@hostb fernando]$ ls -l /home/final
total 16
-rw-rw-r-- 1 503      503          113 Jun 22  2001 dulcinea_report
-rw-rw-r-- 1 fernando fernando    100 Jun 22  2001 panzansan_report
-rw-rw-r-- 1 502      502          88 Jun 22  2001 quixoted_report
[fernando@hostb fernando]$
```

Also, notice that fernando's userid now appears on panzansan's report file. This situation happened because their uids are the same although they are on two different systems. To further clarify, we mean that on HostA, panazansan's uid is 501, and on HostB, fernando's uid is 501. What has happened is the local system (which is HostB) looks to its internal `/etc/passwd` file to determine the uid of the logged-in user. Then, it sees that the file is owned by 501; thus, the system thinks that fernando owns the file. To ensure that this situation does not happen, the system administrator has to manually intervene and make sure that no two uids are the same. Refer now to the following, where we have changed fernando's uid and gid from 501 to 601:

```
[fernando@hostb fernando]$ id
uid=601(fernando) gid=601(fernando) groups=601(fernando),508(reports)
[fernando@hostb fernando]$ ls -l /home/final
total 12
-rw-rw-r-- 1 503      503          113 Jun 22  2001 dulcinea_report
-rw-rw-r-- 1 501      501          100 Jun 22  2001 panzansan_report
-rw-rw-r-- 1 502      502          88 Jun 22  2001 quixoted_report
[fernando@hostb fernando]$
```

Now, panzansan has regained sole ownership of his file, and there is no confusion. Having said all that, we also need to say that if it was your intention to create a userid for dulcinea, panzansan, and quixoted on HostB then

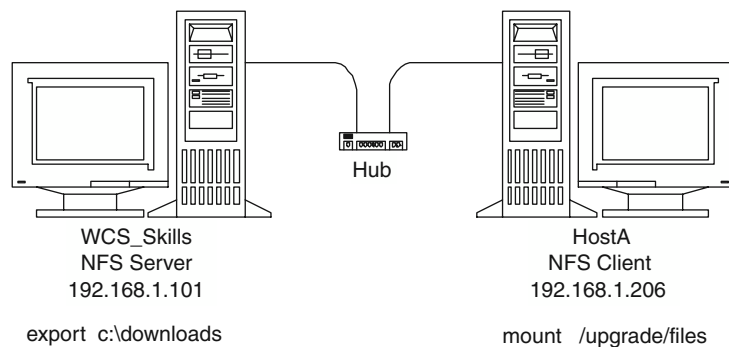
they should have the same uid and gid on both systems. There are tools and utilities like NIS and the SOO environment to help with keeping this kind of data consistent across all servers in large environments.

It should also be noted here that we solved the permission issue by creating a group called reports with a gid of 508 on both HostA and HostB. We make all the users that are going to access this directory a member of this group. Then, group reports was given ownership of the `/home/monthly/reports` directory on HostA. If you go back and review the information, you will see that the one consistent ownership entity between all these users was that they are all members of a group called reports with a gid of 508.

It is our hope that we have demonstrated some of the basic features of NFS and have inspired you to embark on some further research and study. Some good books available explain how to configure all the features of NFS. We recommend that if you are going to use NFS for production purposes, you purchase it. If you are just going to use NFS on a more casual basis, then a thorough reading of the man pages should suffice.

## NFS Solution for Windows

Due to popular demand, we are going to demonstrate NFS sharing between a Windows and Linux system. It has been brought to our attention that many technologists download the files to a Windows system and then have no way of sharing or transferring the files over the network to the Linux box. The example here, illustrated in Figure 4.3, uses a Windows NT Server and a Linux system. There is no NFS capability built into Windows by default like there is for Linux, so we acquired a product called Omni-NFS Enterprise V5.02 from Xlink Technology Inc, [www.xlink.com](http://www.xlink.com).



**Figure 4.3** Windows NFS Server and a Linux NFS client.

We chose this product because it is supported on all Windows platforms and not just on NT. The Xlink people also make a trial version available that will enable you to evaluate the product before purchase.

## Set Up the NFS Server

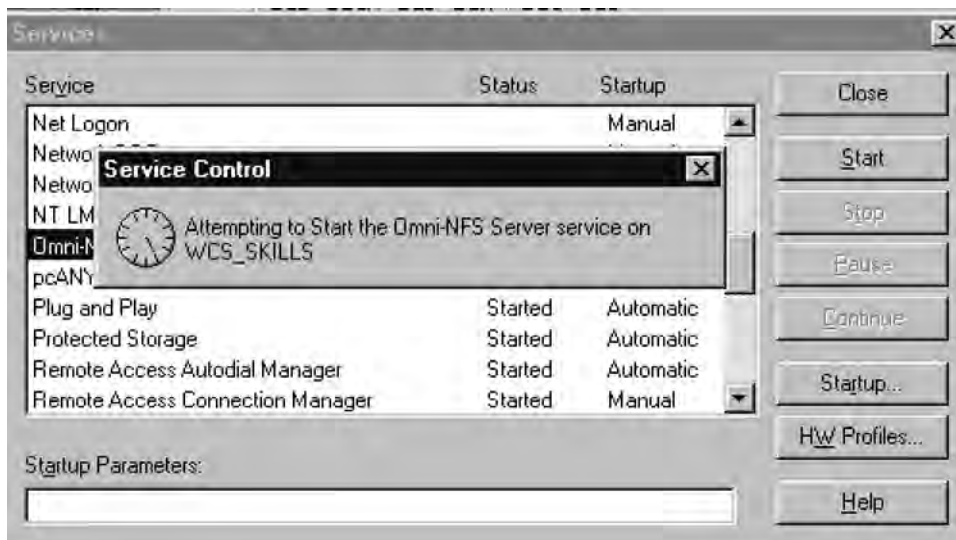
### Step 1. Download and Install NFS Server Code

To demonstrate preparing the NFS server on Windows, we download the trial code from [www.xlink.com](http://www.xlink.com). The downloaded executable file was called *nfse2k.exe*. We accepted all of the defaults during installation. After a successful installation, you should have a new service in the Services panel. Start the Omni-FS Server service first (see Figure 4.4).

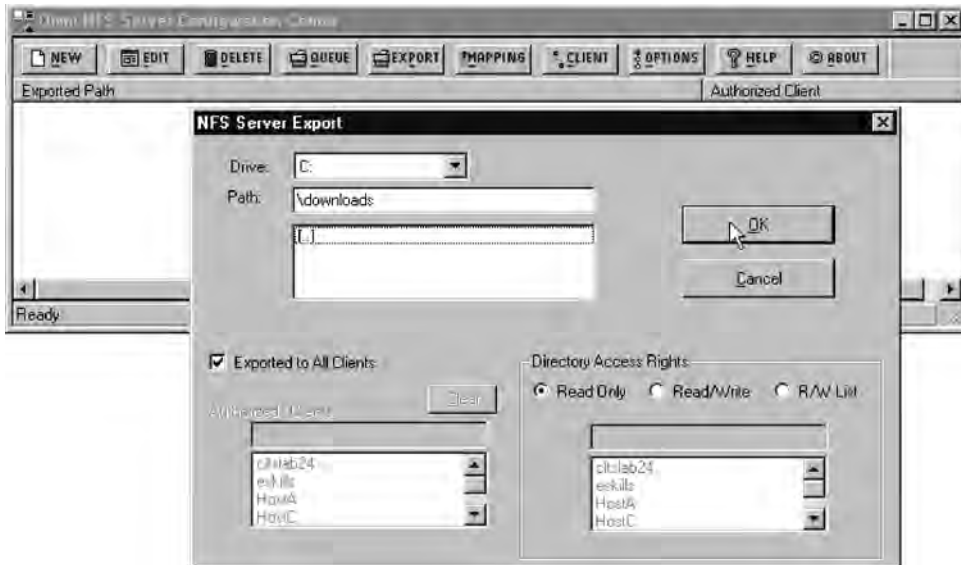
### Step 2. Export the Windows Directory

You can essentially export any directory that you want. We have a directory called *C:\downloads* that we are going to export.

Click Start . . . Programs . . . Omni-NFS Enterprise V5.02 . . . NFS Server. The following panel will appear. Navigate to the directory that you are interested in exporting to the Linux system(s) and click OK. In our example, that is *c:\downloads*. We are going to make this export available to all NFS Clients as Read Only to ensure that no one modifies the files on the Windows system. Make sure that you stop and start the NFS service on the Windows system so that the export takes effect (see Figure 4.5).



**Figure 4.4** Starting the NFS server service on Windows.



**Figure 4.5.** Export the Windows directory by using the server configuration center.

## Set Up the NFS Client

### Step 1. Update /etc/hosts

We are going to configure Linux HostA as a NFS client, but first we should update the hosts file so that HostA can locate our NFS server called WCS\_SKILLS at IP 192.168.1.101:

```
[linda@HostA linda]$ su - root
Password:
[root@HostA /root]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1        localhost.localdomain localhost
192.168.1.206    HostA
192.168.1.101    WCS_SKILLS
```

### Step 2. Create a Mount Point

We decided that we wanted the mounted file system to be `/upgrade/files`. This directory will be an empty directory until the NFS file system is mounted in the next step.

```
[root@HostA /root]# mkdir -p /upgrade/files
[root@HostA /root]# ls -l /upgrade/files
total 0
```

### Step 3. Mount the NFS Exported Directory

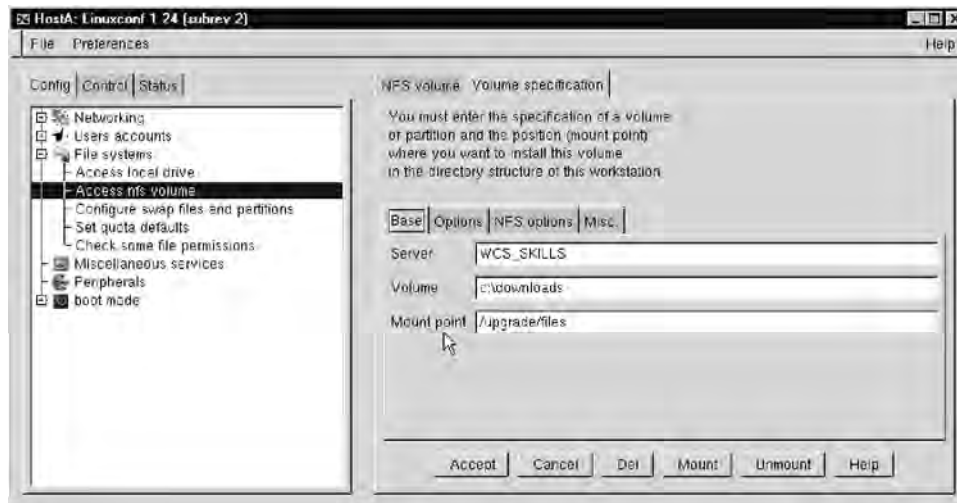
We will mount the NFS exported directory on the Windows system called *c:\downloads* on the newly created mount point called */upgrade/files*. To perform this task, we will use a utility called *linuxconf*. This utility will add an entry to the */etc/fstab* file for the NFS mount. You can edit the */etc/fstab* file directly if you know what the syntax should be.

The *linuxconf* utility is not installed by default, but the *rpm* is supplied on the distribution CDs. Use the *rpm* utility to install *linuxconf*. After installing the *linuxconf* rpm, you also have to start the *linuxconf* service. Use the *ntsysv* utility to start the *linuxconf* service. See Figure 4.6.

### Step 4. Mount the NFS Directory

As we previously mentioned, the *linuxconf* utility did not mount the directory. It only configured the entry in the */etc/fstab* file.

```
[root@HostA /root]# cat /etc/fstab
LABEL=/ / ext2 defaults 1 1
LABEL=/boot /boot ext2 defaults 1 2
/dev/fd0 /mnt/floppy auto noauto,owner 0 0
none /proc proc defaults 0 0
none /dev/pts devpts gid=5,mode=620 0 0
/dev/hda5 swap swap defaults 0 0
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,kudzu,ro 0 0
WCS_SKILLS:c:\downloads /upgrade/files nfs exec,dev,suid,rw 1 1
```



**Figure 4.6** Using *linuxconf* to configure the NFS client mount.

```
[root@HostA /root]# mount /upgrade/files
[root@HostA /root]# ls -l /upgrade/files
total 25913
-rwxrwxrwx    1 4294967294 4294967294 26534489 Jun 26 16:35 linux-
2.4.5.tar.gz
```

We now have all of the files that are in the *c:\downloads* directory on the Windows system available to the Linux system.





# Configuring the Video

In Chapter 3, during the two installation scenarios, we took a shortcut for setting up the monitors. We chose a generic SVGA driver setting, intending at the time to adjust the setup later with the Xconfigurator program. A step or two later, we chose to let the installation program probe for a default screen resolution and color depth. We let the program choose the video mode and color depth and stated that if adjustments had to be made, we would deal with them later as well with Xconfigurator. This chapter is dedicated to the custom video configurations that we postponed in Chapter 3. We discuss the *XF86Config* file, which forms the basis of your X server, and how you can generate and modify that file with the Xconfigurator program to get the most performance out of your video adapter and display. We also briefly discuss another program, SuperProbe, which can help you investigate some of your video system information prior to tackling Xconfigurator.

## Video Signal Generation

---

Before we discuss the video configuration topics, we lay a quasi-technical groundwork for understanding television and video display. Be prepared,

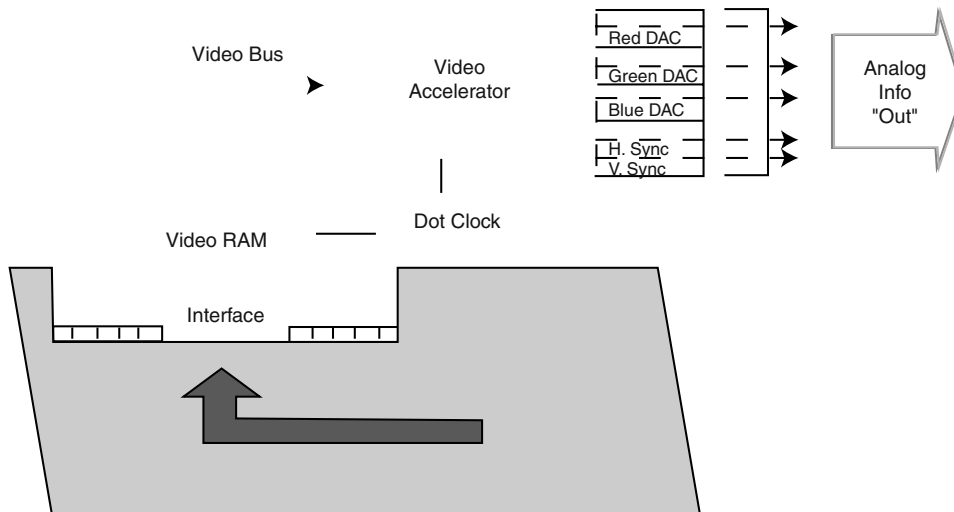


because there are several names that describe the same concepts. Although you might feel that one term or another is better suited, all are legitimate names depending on the situation or context. We focus here only on displays of the *cathode-ray tube* (CRT) variety.

## Video-Adapter Memory

When data are first generated in the *central processing unit* (CPU) of the mainboard (also known as the motherboard), they are formatted as digital signals. The signals that are destined for display on the monitor screen are sent to the video-adapter hardware through one kind of interface or another, depending on which computer system is being employed. The signals might be sent and processed through the mainboard bus itself or through *Peripheral Component Interconnect* (PCI), *Industry Standard Architecture* (ISA), *Accelerated Graphics Port* (AGP), or through other slots or buses. The signals are sent to specifically designed and installed video-adapter cards. In Figure 5.1, we have drawn a very simplified (and separate) video-adapter card.

From the interface, the digital information is likely stored along with the video-hardware drivers in some type of dedicated video memory. The video memory (also known as VRAM or the frame buffer) usually consists of RAM chips that are separate and distinct from those used by the main-



**Figure 5.1** Video-adapter card.

board processor(s). Indeed, the video-memory chips are probably the more advanced (and expensive) dual-port types that enable simultaneous reception and transmission of data. In the case of AGP technology, the video hardware might also have direct access to the computer system's main RAM to supplement the otherwise separate video memory.

Regarding how much video memory you need, the amount required is a function of color depth and resolution. We define these concepts next, mainly with examples.

Looking closely at your monitor screen, you can see thousands of tiny individual dots. The dots are pixels (short for "picture elements") and result from electron beams being focused through the shadow mask onto the interior of the monitor's phosphor-coated screen. Each pixel is composed of three colors: red, green, and blue. Your screen might have as few as 480 lines of 640 pixels or as many as 1,200 lines of 1,600 pixels, depending on your monitor's capacity and the video hardware driving it. Each pixel represents data stored in your video memory.

How much data does each pixel represent? That depends on how many colors you want displayed on the screen. Assume that you need only 16 different colors for very basic spreadsheets, word-processing documents, or other simple images. For 16 colors, each pixel needs four bits (half a byte) of video memory. If you want 256 colors, on the other hand, then each pixel requires eight bits of data (that is, one byte of video memory per pixel). If you want 65,536 colors (for what is commonly called high color), then you need 16 bits (two bytes) of video memory per pixel, and if you want 16.78 million colors (called true color), then each pixel requires 24 bits (three bytes) of memory. There is even a level of color definition based on 32 bits, or four bytes, of video memory per pixel, which could provide up to 4.3 billion colors. These 4-bit, 8-bit, 16-bit, 24-bit, or 32-bit (or, if you prefer, half-byte, one-byte, two-byte, three-byte, or four-byte, respectively) per-pixel specifications are called the color depths.

As we mentioned a minute ago, your screen might consist of as few as 480 lines of 640 pixels or as many as 1,200 lines of 1,600 pixels (or somewhere in between). These dimensions, which are usually described in width in pixels by the number of lines—such as  $640 \times 480$ ,  $1024 \times 768$ ,  $1600 \times 1200$ , and so on—are called the resolution.

Another term you will encounter is *video mode*. Video mode, a description of a monitor's display capacity, is the combination of resolution and color depth. For example, "16 bit,  $800 \times 600$ " is a video mode.

Later in the chapter when you are using Xconfigurator, you will encounter a dialog box titled Select Video Modes. This dialog box presents several mode options, from which you can select one or more based on res-

olution and color depth. Table 5.1 lists the options you will find in the dialog box, plus a few more, and provides estimates of the amount of video memory required to achieve those modes. The bottom line is that if you do not have the required memory, do not expect your combination of video hardware and monitor to give you the mode you request. On the other hand, if you do have a sizable amount of video memory, then you might be surprised at how good and true a picture Linux will enable you to display.

Here is the formula that we used to calculate the video memory required for each video mode in Table 5.1:

$$\text{VRAM required} = \text{color depth} \times \text{line width} \times \text{number of lines} / 1,048,576$$

color depth is bytes per pixel; line width is pixels per line; number of lines is lines per screen, and 1,048,576 is the number of bytes in one megabyte. For example, you would calculate the amount of VRAM required for eight-bit,  $800 \times 600$  video mode as follows:

$$1 \times 800 \times 600 / 1,048,567 = 0.458 \text{ MB}$$

You will probably want to refer to this table later during the Xconfigurator process. Next, remember that your video memory must also contain your video driver software. Consequently, you might not be able to get the resolution and color depth you want from your video memory if the estimated requirement is almost equal to your video memory capacity. Something (resolution or colors, for example) might have to be sacrificed. You might have to learn how to truly custom design your resolution (a process we discuss later).

Although Xconfigurator's Select Video Modes dialog box does not contain a 32-bit color depth option, the XF86Setup program does. For this reason, we included the 32-bit column in the table.

## Basic Video-Adapter Bus

It is likely that your video hardware has separate processors of its own (labeled "video accelerator" in Figure 5.1) that enable the video to be processed independently of the main system processors. The separate processors are often called chipsets as well, or "RAMDAC," because some of them combine video memory and digital-to-analog converters. Xconfigurator and other utilities might refer to the processors by using these terms. During the video configuration exercise at the end of this chapter, Xconfigurator will suggest that you consult beforehand some of the video-server manual

**Table 5.1** Video Modes

RESOLUTION (PIXELS BY LINES)	COLOR DEPTH 8 BITS	COLOR DEPTH 16 BITS	COLOR DEPTH 24 BITS	COLOR DEPTH 32 BITS
	(1 BYTE/PIXEL, OR 256 COLORS)	(2 BYTES/PIXEL, OR 65,536 COLORS)	(3 BYTES/PIXEL, OR 16.78 MILLION COLORS)	(4 BYTES/PIXEL, OR 4.29 BILLION COLORS)
640 × 480	0.293MB	0.586MB	0.897MB	1.172MB
800 × 600	0.458MB	0.916MB	1.374MB	1.831MB
1024 × 768	0.750MB	1.500MB	2.250MB	3.000MB
1152 × 864	0.950MB	1.899MB	2.848MB	3.797MB
1280 × 1024	1.250MB	2.500MB	3.750MB	5.000MB
1600 × 1200	1.832MB	3.663MB	5.493MB	7.325MB

pages and chipset-specific READMEs. It will also suggest that you know beforehand the name of the chipset and the amount of video memory on your video card.

How do you estimate the amount of work that video processors perform? Earlier, we discussed the memory required to store the information that will be displayed. The video processors have to move the stored data through the video bus and to the monitor and move it there according to the *vertical synchronization frequency* (VSF, also known as the screen refresh rate). We discuss VSF in a bit more detail later in the chapter. The VSF is the rate per second at which the monitor's displays are redrawn. The range is from 40 to 160 screen refreshes per second (measured as cycles per second, or hertz, and abbreviated as Hz). To quickly summarize, the work (measured in megabytes per second) is approximately equal to the amount of data per screen multiplied by the number of screens drawn per second. Table 5.2 provides estimates of the image-data transfer rate by the video processors, using example refresh rates of 50, 60, and 75 Hz.

To calculate these estimates, we used the following formula:

```
Data transfer rate = line width x number of lines x color depth x  
refresh rate / 1,048,576
```

line width is pixels per line; number of lines is lines per screen; color depth is bytes per pixel; and 1,048,576 is the number of bytes in one megabyte.

For example, you would calculate the data transfer rate for  $800 \times 600$  resolution, 24-bit color depth, and a 60 Hz refresh rate as follows:

```
800 x 600 x 3 x 60 / 1,048,576 = 82.4 MB/second
```

Note that Table 5.2's heading is Image-Data Transfer Rate. The visible image is not the only data processed and transferred from the video hardware to the monitor. Later in the chapter, we discuss additional information that must be transferred—the synchronization signals that govern the deflection of the electron beams. These signals add even more data. The *Video Electronics Standards Association* (VESA), for the sake of safety, recommends that you always add 4 percent to 5 percent to any line-rate calculation (the number of lines from a given resolution multiplied by the desired or given refresh rate) because of the overscan sync signal. There is also another 25 percent or so of overhead for the retrace of horizontal scan lines. We discuss and clarify these terms and concepts as we progress through the chapter.

**Table 5.2** Image-Data Transfer Rate

RESOLUTION	COLOR DEPTH 8-BIT			COLOR DEPTH 16-BIT			COLOR DEPTH 24-BIT			COLOR DEPTH 32-BIT		
	REFRESH RATE			REFRESH RATE			REFRESH RATE			REFRESH RATE		
	50 HZ	60 HZ	75 HZ	50 HZ	60 HZ	75 HZ	50 HZ	60 HZ	75 HZ	50 HZ	60 HZ	75 HZ
640 × 480	14.7	17.6	22.0	29.3	35.2	44.0	44.0	52.7	65.9	58.6	70.3	87.9
800 × 600	22.9	27.6	34.3	45.8	54.9	68.7	68.7	82.4	103.0	91.6	109.9	137.3
1024 × 768	37.5	45.0	56.3	75.0	90.0	112.5	112.5	135.0	168.8	150.0	180.0	225.0
1152 × 864	47.5	57.0	71.2	94.9	113.9	142.4	142.4	170.9	213.4	189.8	227.8	284.8
1280 × 1024	62.5	75.0	93.8	125.0	150.0	187.5	187.5	225.0	281.3	250.0	300.0	375.0
1600 × 1200	91.6	109.9	137.3	183.1	219.7	274.7	274.7	329.6	412.0	366.2	439.5	549.3

Note: The image-data transfer rate is measured in megabytes per second.

The video processors are governed by the video hardware's on-board memory clock (also called the dot clock or VGA memory clock), which performs the same function as the mainboard bus clock but is dedicated to the video bus alone. The activity of the on-board memory clock, which we refer to as the dot clock henceforth, is measured in millions of cycles per second (megahertz, or MHz). Meanwhile, the video hardware, especially if it is located on a separate adapter card, also probably has its own separately sized bus (likely 32- or 64-bit) to optimize the processing and movement of data through the video adapter to the monitor.

Thus, the video signals are processed—and after leaving the accelerator, the actual picture element signals are converted from digital to analog format in the three (one for each of the primary colors: red, green, and blue) *digital-to-analog converters* (DACs) on the card (called A/Ds in days gone by). Why are they converted? Although this story is long, the fundamental reason is to provide more colors and improved control of light intensity on the screen.

Only five signals leaving the video adapter are represented in Figure 5.1. In reality, however, there are more. Nevertheless, the five we include in the illustration are the most important at this stage in the discussion: the red, blue, and green signals plus the two synchronization signals. We discuss the importance of all the signals in the next section.

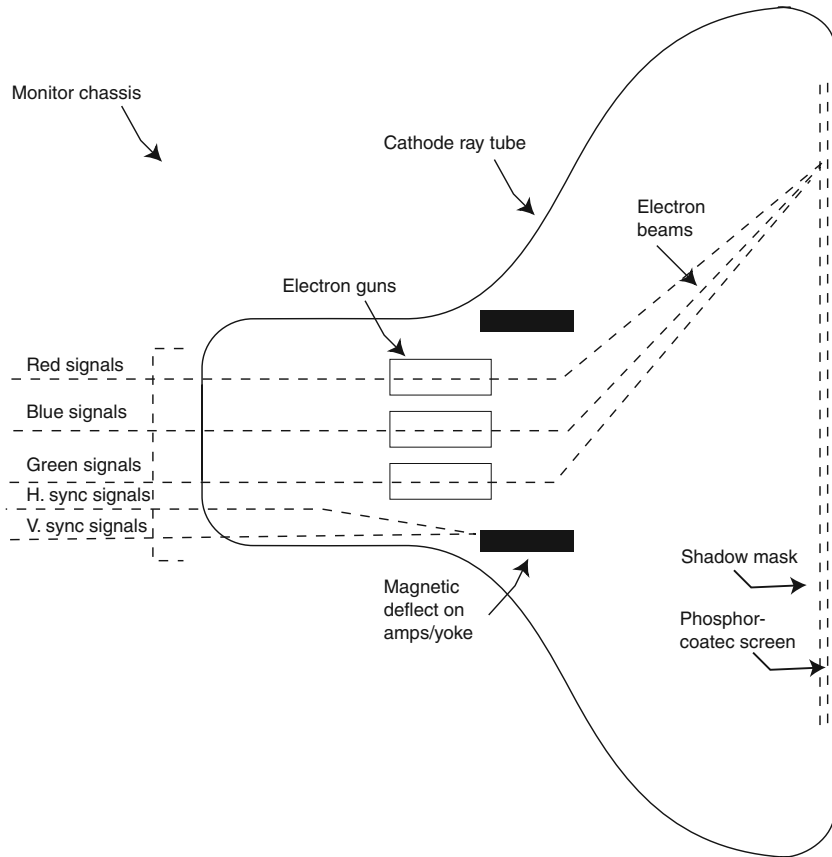
## **Video Display**

---

If your video configuration is not behaving properly and you are getting horrible lines, blinking and blank screen failures it is because the monitor capabilities are not defined correctly. A basic understanding of monitor technology will help you effectively troubleshoot most video problems. It would be nice if there were only one kind of video technology out there in the market. But such is not the case.

### **Basic Monitor Design**

As illustrated in Figure 5.2, within the monitor the color signals (which are constantly varying voltages) are amplified and enter respective red, blue, and green electron guns. The variations of the voltages govern the intensity of the electron beams that are emitted from those guns. Meanwhile, the synchronization signals are also amplified and then enter the magnetic deflection yoke (a collection of deflection coils), wherein they govern the aiming and movement of the electron beams as well as the size of the indi-



**Figure 5.2** A simplified model of video-display design.

vidual picture elements (that is, the pixels). The impact of the electrons on the phosphor-coated screen creates the pixels (and thus, the display image).

The screen is coated in stripes of phosphor material (phosphor is roughly defined as a material that glows when energized by electrons). Each stripe glows a different color—red, green, or blue—when energized. The stripes might be oriented vertically (as in Sony’s Trinitron) or quasi-horizontally (where the three colors are arranged in a triangle). In the quasi-horizontal orientation, the left end of the stripe is higher than the right end.

The shadow mask, next to the phosphor-coated screen, keeps the red, green, and blue electron beams precisely on target within their respective color stripes on the screen. Depending on the nature of the monitor’s technology, the shadow mask might have vertical slots, circular holes, elliptical holes, and so on.



The electron guns are a combination of connected high-voltage cathodes (which emit electrons) and anodes. The electron beams are also called cathode rays, which is why monitors have historically been called cathode ray tubes. In fact, a part of the monitor is actually called the CRT (on a TV, it is called the picture tube). The cathode ray tube is made of thick glass, and the pressure within it is less than atmospheric (that is, it is under a vacuum).

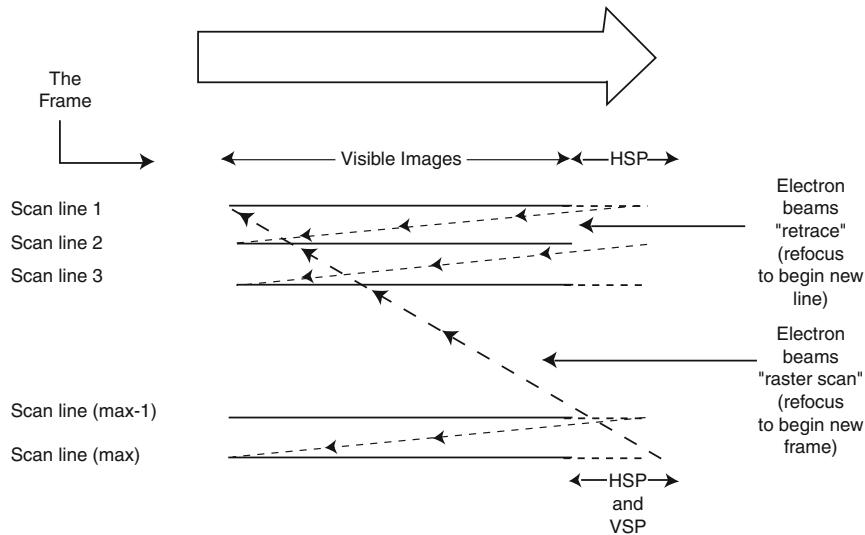
If you wish to do further research on monitor design, check any elementary radio and television electronics textbook or manual.

## Video Frame and Synchronization

In the three previous sections, we discussed how display signals are generated and sent to the monitor and how electrons are emitted from guns and then impact on a phosphor-coated screen to create a display image. In this section, we discuss the movement of the electron beams across and down the screen itself.

The entire display image, from the top-left dot to the bottom-right dot and including all of the lines in between, is called a frame. (When we discuss the concept of interlacing, you will see how a frame is also considered to be composed of two fields.) To create a frame, the electron beams are first aimed at the top-left corner of the screen (as seen by the system user). This action is called “beginning a frame.” The deflection yoke then causes the beams to traverse horizontally from left to right straight across the screen, creating a scan line. For most of the travel across the screen, the electron beams hit and activate tiny phosphor dots (because the phosphor material coats the screen in very narrow stripes, according to the colors they will produce when hit), causing them to light up. Figure 5.3 illustrates how a frame is created.

The signals (analog voltages) received at the electron guns also determine the intensity of the light emitted by the phosphors. After a beam leaves a phosphor, the phosphor continues to glow very briefly. The capability of a phosphor to continue to glow after electron impact is called its persistence. (The phosphors would eventually fade, but they are re-energized on the next return sweep of the beams—but we are getting ahead of ourselves a little.) If the phosphor persistence is too long, you will see ghost images on the screen because previous images will not have faded sufficiently before the phosphors are re-energized. If persistence is too short, the screen will be perceived to flicker; that is, to brighten during electron impact, darken immediately, brighten significantly during the next electron impact, then darken, and so on. Screen flicker is not only annoying but can also cause severe discomfort.



**Figure 5.3** Creating a frame.

Near the end of each horizontal scan line, the beams from the electron guns are actually shut off very briefly just before a *horizontal synchronization pulse* (HSP) is received by the deflection yoke from the video adapter. The guns are then retraced back to the left edge of the screen, except that the deflection yoke alters the angle of the guns so that when electron-beam emission is resumed, the beams will now focus on the next-lower line of the display (when we discuss interlacing, you will see the exception to this rule).

The rate of the sweeping motion of the electron beams from left to right, line by line, is called the horizontal synchronization frequency, the *horizontal scan frequency* (HSF), or the line rate, and this value is measured in kilohertz (thousand cycles per second, or KHz; 1 Hz equals one cycle per second). In other words, the horizontal scan frequency is the number of times per second that the monitor's electron guns can write a horizontal scan line. The actual calculation is the number of lines drawn per screen by the number of screens drawn per second. Customarily, HSFs are in the 30 to 70 KHz range.

The electron beams will continue to traverse from left to right across the display and then be returned to the left side of the next-lower line until they reach the bottom line of the display. Again, they will travel horizontally until they approach the right side. Near the end of the very last line, however, the deflection yoke will receive a slightly different signal, a *vertical synchronization pulse* (VSP), from the video adapter. The VSP will tell it to shut off the electron beams and this time focus the guns back at the left side

of the display's first line. Thus, the beams end the frame, and then raster scan (that is, travel back to the upper-left corner) to begin a new frame.

The number of times per second that the electron beams move from the top-left dot of the frame to the bottom-right dot is called the vertical synchronization frequency or *vertical scan frequency* (VSF). The VSF is also measured in hertz (Hz), and the customary range is from 50 to 100 Hz (although the lowest can be 40 Hz and the highest more than 160 Hz).

The VSF is also called the vertical refresh rate or simply "refresh rate." Remember the previous mention of refresh rate. We often generalize and say that a screen is refreshed about 60 times a second. Thus, the basic meaning of all these vertical sync or refresh terms are simply the number of times the electron beams redraw the display on the screen. You might notice that if the refresh rate drops below 50 Hz, screen flicker can result due to the lack of sufficient phosphor persistence. As we mentioned earlier, screen flicker should be avoided because it can cause discomfort to the user.

Let's quickly review video frame and synchronization basics: HSFs are in the 30 to 70 KHz range while VSFs are in the 40 to 160 Hz range. For every VSP (once per frame), the beams have to experience many hundreds, or even thousands, of HSPs (once per line). VSFs and HSFs are critical in designing the amplifiers used by the magnetic deflection yoke.

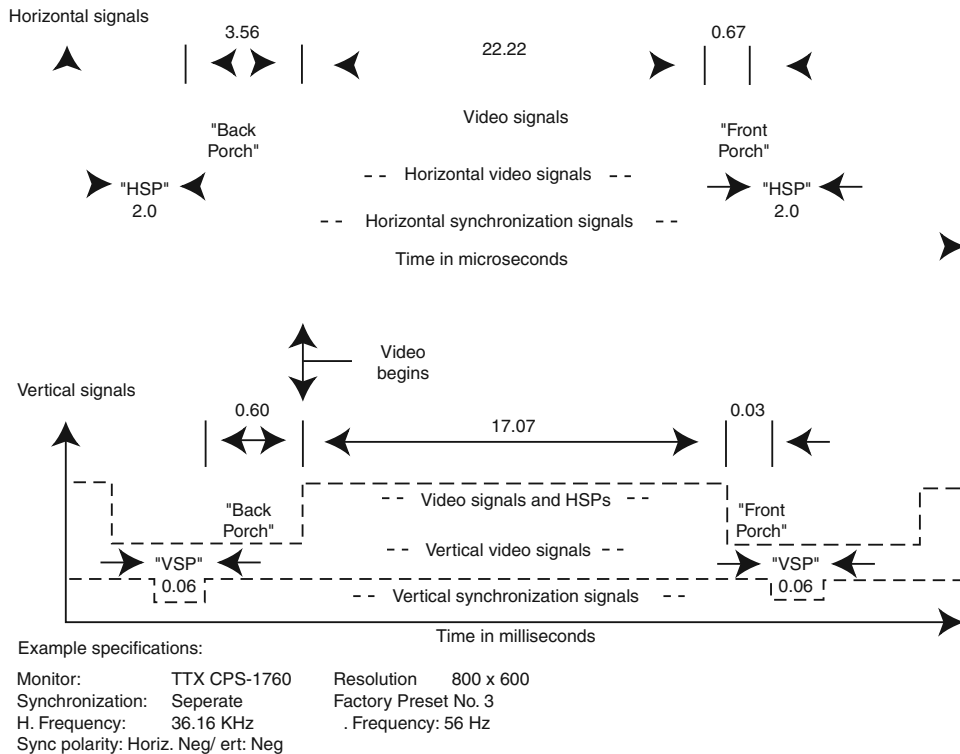
Here is another definition worth mentioning. The time it takes to stabilize and synchronize the vertical and horizontal movements is also referred to as sync pulsing. In the next section, we again describe raster scanning, but we explore the actual time taken and where the pulses travel.

Without a doubt, you have noticed that these terms and specifications can be easily confused. Take the time to understand them and their potential impacts on the configuration of your video display. The Linux-provided capability to specify and modify can be a great boon to optimizing your equipment. Misunderstanding and misusing the concepts can lead to frustration, however, and occasionally to the destruction of your equipment.

## How HSPs and VSPs Are Related

In the previous section, we illustrated a typical raster scan of the visual image on a monitor screen. We mentioned that HSPs occur at the end of each line and that VSPs occur at the end of each frame.

Figure 5.4 shows two diagrams, each consisting of basic voltage versus time graphs for video signals and synchronization pulses. The diagrams are typical monitor timing charts provided by manufacturers. The graphs



**Figure 5.4** Voltage versus time graphs for video signals and synchronization pulses (diagrams are not to scale).

in the figure are from a TTX CPS-1760 17-inch monitor at  $800 \times 600$  resolution. Note that the timing numbers change for different resolutions.

### Horizontal Signals

Note that in the horizontal signals diagram in Figure 5.4, time is measured in microseconds (millionths of a second). Now, look at the left side of the horizontal synchronization signals graph, which shows the synchronization signals that travel from the video adapter to the magnetic deflection amplifiers in the monitor along the dedicated horizontal signals line. The first event is a two-microsecond HSP, which has occurred at the right end of a horizontal scan line. Note that the HSP signal appears to be negative with respect to the normal synchronization signal, which is the specification for this monitor at this resolution. When a multiple-frequency monitor like this one detects this signal orientation and then reads the orientation

(positive or negative) and duration of the vertical synchronization signal, it is tipped off as to what resolution and frequencies to expect.

Looking at the video signals graph immediately above that point, you see that the next event is a 3.56-microsecond back porch, which is the travel time for the electron beams to go from the right side of one line to the left side of the next lower line (during which you see that the electron guns are turned off). Immediately after that back porch period, the electron beams are activated as analog video signals and are sent from the video adapter to the electron guns through the dedicated red, green, and blue signal lines. For  $800 \times 600$  resolution, the video signals (which are really three sinusoidal, or wavy, signals but are simplified as one flat voltage for these timing charts) are sent for 22.22 microseconds, after which the electron guns are deactivated for a 0.67-microsecond front porch period. At the end of the front-porch period, you glance down again at the horizontal signals graph and see that another HSP comes through, which ends the scan line and begins the cycle anew.

You can see that for the total 22.22 microseconds, during which the image data were sent, there has been an overhead of 6.23 microseconds of other signal activity. So, each scan line took 28.45 microseconds.

### ***Vertical Signals***

In the vertical signals diagram, note first that time here is measured in milliseconds (thousandths of a second), not microseconds. A millisecond is equal to 1,000 microseconds. Now, let's start at the left side of the vertical synchronization graph, which shows the synchronization signals that travel from the video adapter to deflection amplifiers in the monitor along their own dedicated vertical synchronization line. Here, you see that the first event is a 0.06-millisecond VSP, which is the signal received at the bottom of a frame. Looking at the video signals graph immediately above the VSP, you see a 0.60-millisecond back porch, which is the travel time for the electron beams to go from the bottom-right corner of the frame to the top-left corner, during which the electron beams are deactivated. Immediately after that back porch period, the electron beams are reactivated at a time that coincides exactly with the activation on the first horizontal scan line. Following along the video signals and HSP's portion of the vertical video signals, you see that for the next 17.07 milliseconds, all the horizontal scan lines are drawn on the screen. If you divide 17.07 milliseconds by the total 28.45 microseconds per scan line, you will get exactly 600 scan lines, which is an exact match for the  $800 \times 600$  resolution specification. After the video signals and HSP period, there is an additional 0.03-millisecond front-porch

period during which the electron beams are deactivated as a prelude to the next VSP, which will begin the cycle anew.

Note that there is overhead in the vertical signal cycle, just as there was for the horizontal signal cycle. This time, the vertical overhead is 0.69 milliseconds versus the 17.07 milliseconds of scan lines. The overhead is roughly equal to scanning an additional 23 or so lines (that is, an additional 4 percent, roughly), which is why it is often called overscan. We mentioned before that the VESA recommends that you always add 4 percent to 5 percent to any line-rate calculation because of the overscan sync signal. Overscan is mentioned again in the next section on monitor capabilities.

Earlier, we discussed resolution, color depth, and video memory. Now that we have discussed horizontal and vertical synchronization, let's take a brief look at how the video configuration file *XF86Config* stores resolution information in modelines. The following is a typical modeline:

```
Modeline "800 × 600" 69.65 800 864 928 1088 600 604 610 640 -HSync -VSync
```

The numbers and other codes in this modeline are described in Table 5.3.

Despite what we discussed regarding timing charts for monitors, you can see that the *XF86Config* file stores video-mode timing information according to a pixel and line convention. Note that all horizontal scan-line numbers are divisible by eight, indicating the discrete bytes of information. The vertical numbers need not be divisible by eight, however, because all the lines are already broken into bytes. In addition, the file treats the beginning of the visible display as zero for all counting.

Modes can be specified in a multi-line format. The following is equivalent to the earlier modeline example:

```
# Mode "800 × 600"
# DotClock 69.65
# Htimings 800 864 928 1088
# Vtimings 600 604 610 640
# Flags "-HSync" "-VSync"
# EndMode
```

## Monitor Performance and Capabilities

Earlier, we discussed the image-data transfer rates that required color depths and display resolutions, followed by the basic operations of a monitor. The impression we might have left with you thus far is that your monitor can always handle anything you send to it if you have sufficient

**Table 5.3** Modeline Numbers and Codes

NUMBER/CODE	DESCRIPTION
800 × 600	Resolution
69.65	Dot Clock Frequency, in MHz
800	Pixel count at the end of the visible part of the horizontal scan line and the beginning of the HSP's front porch
864	Pixel count at the end of the front porch and the beginning of the HSP itself
928	Pixel count at the end of the HSP and the beginning of the HSP's back porch
1088	Pixel count at the end of the back porch
600	Line count at the bottom of the visible part of the screen and the beginning of the VSP's front porch
604	Line count at the end of this front porch and the beginning of the VSP itself
610	Line count at the end of the VSP and the beginning of the VSP's back porch
640	Line count at the end of the VSP's back porch
-HSync	HSP will be a negative polarity with respect to its normal voltage level
-VSync	VSP will be a negative polarity with respect to its normal voltage level

video-adaptor hardware. This impression might or might not be accurate. This issue is a bit tricky. Several factors can affect a monitor's performance and capabilities:

- Monitor age
- Bandwidth
- Vertical frequency
- Horizontal frequency
- Dot pitch

Let's consider each point individually.

## Monitor Age

If the monitor is a new multiple-frequency device, then it might be capable of handling anything you send to it—especially if the video-adaptor hardware and computer system predate it by a year or so. If you are trying to make use of an older and therefore less-expensive monitor, however, then you should get a handle on what you *want* the monitor to do for you versus what it *can* do for you. It could become a performance bottleneck, or worse, a bit of a bomb (literally).

## Bandwidth

A monitor's bandwidth is a measure of its capability to accommodate the information sent to it. If the monitor is a new model, it might reflect the modern manufacturer's expectations: namely, that video-adaptor hardware will continue to become more powerful (that is, capable of sending more information via the DCF, or dot clock frequency), so monitors should be manufactured to anticipate DCF increases. That is why with some systems people say, "Don't worry, it's a new monitor, so the DCF probably don't tax the monitor's bandwidth."

As we said earlier, though, if the monitor is older—perhaps a fixed-frequency color model but with a large screen size that could be potentially beneficial to your operations—then you have to determine its capabilities as best you can.

In this book, we call the bandwidth output from the video adapter the DCF and the bandwidth capability of the monitor the bandwidth. Both are measured in MHz. The higher the DCF, the more information being transmitted to the monitor in the form of refreshes per second, resolution, color depth, and line rates. The higher the bandwidth, the higher the monitor's capabilities to cope with the DCF being sent to it.

If you do not know the dot clock frequency of your video-adaptor hardware, the XF86Free - X Server might be able to help. Try running the `startx` script and then exit and read the output. Even if it fails, it should give you some useful information in the standard error returned. Example output from the script follows:

```
[root@hostname /home]# cat versa-video
(-- ) SVGA: CHIPS: probed memory clock of 31499 KHz
(-- ) SVGA: chipset: ct65550
(-- ) SVGA: videoram: 2048k
(**) SVGA: Using 16 bpp, Depth 16, Color weight: 565
```



```
(--) SVGA: Maximum allowed dot-clock: 44.098 MHz
(**) SVGA: Mode "800 X 600": mode clock = 40.000
(--) SVGA: Virtual resolution set to 800 X 600
(--) SVGA: SpeedUp code selection modified because virtualX != 1024
(--) SVGA: CHIPS: 1098752 bytes off-screen memory available
(--) SVGA: CHIPS: SpeedUps selected (Flags=0xF)
(--) SVGA: CHIPS: Memory mapped I/O selected
(--) SVGA: Using XAA (XFree86 Acceleration Architecture)
(--) SVGA: XAA: Solid filled rectangles
(--) SVGA: XAA: Screen-to-screen copy
(--) SVGA: XAA: 8x8 pattern fill
(--) SVGA: XAA: 8x8 color expand pattern fill
(--) SVGA: XAA: CPU to screen color expansion (bitmap, TE/NonTE
imagetext, TE/NonTE polytext)
(--) SVGA: XAA: Using 6 128x128 areas for pixmap caching
(--) SVGA: XAA: Caching tiles and stipples
(--) SVGA: XAA: Horizontal and vertical lines and segments
waiting for X server to shut down
(**) Supplied
(--) Probed
```

Thus, we have a maximum allowed dot clock of 44.098 MHz. XconfiguratorY will not ask you for this information, but it is still good to know. Remember, though, that a monitor must exceed the 44.098 MHz number because it will have to cope with retrace and overscan signals.

If you do not know the bandwidth of your monitor, consult Table 5.4, which indicates the bandwidth that a monitor must have to cope with the respective image-data transfer rates (plus retrace and overscan).

To calculate these estimates, we used the following formula:

$$\text{Bandwidth} = \text{line width} \times \text{number of lines} \times \text{color depth} \times \text{refresh rate} \times 1.045 \times 1.25$$

line width is pixels per line, number of lines is lines per screen, color depth is bytes per pixel, refresh rate is screen redraws per second, 1.045 is the number used to compensate for overscan, and 1.25 is the number used to compensate for retrace.

For example, you would calculate the bandwidth necessary for 800 × 600 resolution, 24-bit color depth, and 60 Hz refresh rate as follows:

$$800 \times 600 \times 3 \times 60 \times 1.045 \times 1.25 = 113 \text{ MHz}$$

**Table 5.4** Monitor Bandwidth Requirements

RESOLUTION	COLOR DEPTH = 8 BIT			COLOR DEPTH = 16 BIT			COLOR DEPTH = 24 BIT			COLOR DEPTH = 32 BIT		
	50 HZ	60 HZ	75 HZ	50 HZ	60 HZ	75 HZ	50 HZ	60 HZ	75 HZ	50 HZ	60 HZ	75 HZ
640 × 480	20.1	24.1	30.1	40.1	48.15	60.2	60.2	60.2	72.2	80.3	96.3	120
800 × 600	31.4	37.6	47	62.7	75.24	94.1	94.1	94.1	113	141	150	188
1024 × 768	51.4	61.6	77	103	123.3	154	154	154	185	231	247	308
1152 × 864	65	78	97.5	130	156	195	195	195	234	293	312	390
1280 × 1024	85.6	103	128	171	205.5	257	257	257	308	385	411	514
1600 × 1200	125	150	188	251	301	376	376	376	451	564	602	752

Note: Monitor bandwidth is measured in MHz; estimates assume 30 percent extra for overscan and retrace.

## Vertical Frequency

The maximum vertical frequency (or maximum refresh rate) sent from the video-adaptor hardware to a monitor must not exceed the monitor's own vertical frequency.

## Horizontal Frequency

The line rate received by a monitor must not exceed the monitor's horizontal frequency. Remember, the line rate is equal to the number of HSPs, which in turn is equal to the number of lines in a desired or given resolution, such as the 600 in  $800 \times 600$  multiplied by the desired or given refresh rate, such as the 60 in 60 Hz.

In previous sections, we mentioned that VESA, for the sake of safety, recommends that we always add 4 percent to 5 percent to any line-rate calculation because of the overscan sync signal. This situation is less likely to be a problem at the present ("present" being defined as from the year 2000 and into the future), because there are more and more multiple-frequency monitors being used. Multiple-frequency monitors can detect the incoming horizontal and vertical frequencies and adjust their circuits so that their amplifiers can handle them. The ways in which these monitors work are summarized as follows:

**Through combinations of synchronization polarities.** For example, the original VGA cards used this technique to tell the monitors whether to create screens with 350, 400, or 480 lines.

**Through the length of the vertical synchronization pulse.** The monitors measure the length of the pulse, convert it to the equivalent number of scan lines that could be drawn during that time, and then calculate the corresponding screen length and width.

**By using their own memory and microprocessors to analyze synchronization signals** (frequencies, timings, and the like) and then calculating and setting up the corresponding resolutions.

## Dot Pitch

Monitor size is a very important issue. You should ask yourself, "How big a monitor do I need for the resolution(s) I want?" Unfortunately, this question is not simple to answer. It depends on the monitor capabilities we have described thus far, plus a grasp of the dot pitch of the monitor. What is the dot pitch? It is the distance on your monitor, generally measured in

millimeters, between successive phosphor stripes of the same color. Dot pitches generally vary between 0.31 and 0.22 millimeters.

On any screen, the dot pitch is easy to see if you use a magnifying glass. On something like a Sony Trinitron screen, it is especially easy to see because the phosphor stripes are vertical and the shadow mask (which Sony calls its “aperture grille”) is slotted so that it does not cover too much of the phosphor area. That said, we do not suggest that you measure the dot-pitch distance yourself. It is a lot easier to consult the manufacturer’s specifications in the monitor’s manual.

You might notice that on older color screens, seeing the dot pitch is a little more difficult because the shadow masks have holes, as opposed to slots, and the color stripes are slanted left to right. The dot pitch, therefore, is not measured strictly vertically or horizontally, but rather inclined. Your manufacturer’s manual will specify horizontal and vertical measurements, but to get the true measurements, the manufacturer will have applied a correction factor to account for the angle of the phosphor stripes. While we do not recommend trying to measure the dot pitch on your monitor, it is worthwhile to take a close look to determine the type of monitor screen technology you have.

Why is dot pitch important? If you are using a monitor of a certain size and you want to obtain a certain resolution on it, then you have to be confident that your monitor’s dot pitch is small enough to show that resolution. Let’s use the Sony CPD-120VS 15-inch monitor as an example. The manual says that it has an aperture grille (that is, dot pitch) of 0.25 mm. Assume that you want to display the monitor’s maximum-rated resolution of 1,280 dots by 1,024 lines. First, how big is that image? A dot pitch of 0.25 mm means that there are 101.6 dots per inch. If you divide the 1,280 dots by 101.6 dots per inch, you get an image with a width of 12.6 inches (or 320 mm). Consulting the manual, we see that the screen is only 11.2 inches (284 mm) wide. If the math does not work out perfectly, read on.

At this juncture, all we can say is you have to *be careful of what you read in the manufacturer’s manual*. It turns out that the dot pitch is correct. Because the manual was written for both the Sony CPD-120VS (15-inch) monitor and the CPD-220VS (17-inch) monitor, the maximum resolution for the 17-inch monitor is reproduced in the 15-inch monitor manual. The viewable image size is included for both monitors, however, and the 15-inch monitor has a viewable image size of 284 mm. If you divide 284 mm by the dot pitch of 0.25 mm, you get a pixel width of 1,136. Thus, the maximum *standard* size of your display is likely  $1024 \times 768$ .

Conversely, if you were adamant about displaying 1280x1024 on a 15-inch monitor, you would have needed a dot pitch of 0.22 mm (284

mm/1280 pixels = 0.22 mm). This situation is impossible for the Sony CPD-120VS, unfortunately, and we could not find a single product in the monitor specification database that could do it, either. The 0.22-mm dot pitch seems only to be available in 21-inch monitors as of this writing. The 17-inch Sony CPD-220VS monitor might actually handle 1280x1024, although the extra time required for horizontal sync pulses and their respective front porch and back porch timings might not enable it. Again, you have to be careful with what the manufacturer tells you in manuals. Your best bet is to check out the resolutions that can be displayed on a monitor prior to making your purchase. In the meantime, Table 5.5 might be useful. Remember, the numbers in the table might not be totally reliable because there might be additional signal overhead (overscan and retrace) or monitor limitations (dot pitch) that prevent these resolutions from being achieved.

There are several good sources of additional information available, but none of them provides a comprehensive review of all aspects of image generation and display. One interesting source is Eric S. Raymond's "XFree86 Video Timings HOWTO," wherein he describes how to customize the size of a monitor's display. It is handy and provides a detailed illustration of why the various screen widths must accommodate the actual size of information bytes. He also presents a small table of minimum bandwidth requirements per resolution.

You can also access <http://ms.ha.md.us/~hawks/hardware/monitor.html> for a table of monitor specifications, including bandwidth, dot pitch, display size, synchronization specifications, and so forth. The only handy column missing in that table is nominal resolution size, but you can quickly calculate this value from the given display sizes. Another reasonably good discussion of video generation and display is connected to the table. Other good Internet sites are [www.pctechguide.com](http://www.pctechguide.com) and [www.zdnet.com](http://www.zdnet.com).

**Table 5.5** Maximum Resolution for Standard-Size Monitors

NOMINAL MONITOR SIZE	MAXIMUM RESOLUTION
14"	800 × 600
15"	1024 × 768; 800 × 600
17"	1024 × 768
19"	1280 × 1024
21"	1600 × 1200

### ***The Bottom Line: Match Your Video Hardware and Monitor to Your Expectations***

If you have expectations of your monitor (for example, you might want a huge monitor to handle high resolutions so that you can have several programs open and displayed at once), then try to ensure that your video hardware can process and send the necessary information to your monitor. You will need adequate video memory and a bus sufficient to handle the flow. Then, on the other end, try to ensure that your monitor will indeed handle everything that the video hardware sends. Its bandwidth has to be high enough to handle all the information, and its sync frequencies must be high enough to handle line rates and refresh rates. It would be a shame to have video hardware that is capable but too powerful for the monitor or a monitor that is choked back to lower resolutions because the video adapter cannot send it information quickly enough. Most importantly, it would be a shame if you could not do the work you need to do without spending more money.

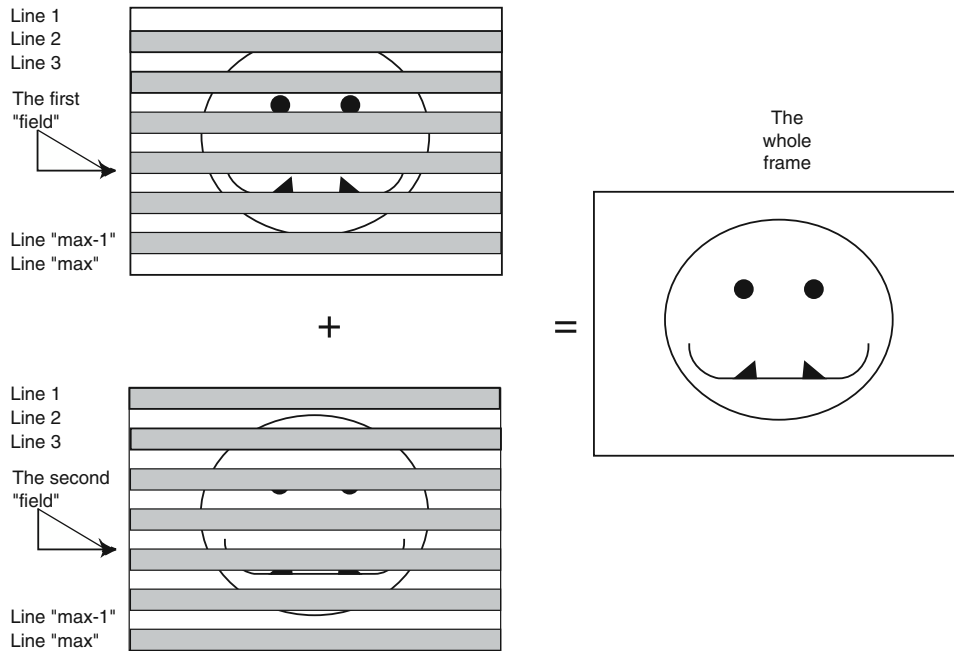
In the next section, we discuss a technique for getting high resolution out of a lower-end monitor; that is, a monitor with bandwidth that might not permit it to support high DCFs.

## **Interlacing**

---

In previous sections, we examined how the electron beams create frames line by line. Some video adapters, however, send their information so that the beams scan only every second line on one pass (that is, the odd-numbered lines) and then go back for a second pass, where they scan the lines they missed on the first pass (that is, the even-numbered lines). This process is called interlacing, and each full pass is called a field. Thus, two complete fields will equal one frame. Figure 5.5 illustrates this technique.

Why would you want to use interlacing? Interlacing can help you by reducing the amount of new information carried with each refresh while maintaining high resolution. This function would be useful, for example, if your monitor had only a certain bandwidth capacity or if your video-adaptor hardware had a limited DCF such that you could not view a high resolution without flicker or without getting an error message announcing that you had exceeded bandwidth limits. Thus, using an interlaced mode might allow you to display higher resolutions without overtaxing the



**Figure 5.5** A simplified illustration of interlacing.

video adapter or monitor amplifiers, or you might obtain a higher resolution on a monitor that has less-expensive components (amplifiers, processors, and the like) or that is based on an older technology.

Let's assume that you are trying to display a high resolution with respectable color depth, but your resulting refresh rate comes out low (say, at 45 Hz). The flicker could be most annoying. In this case, you could try an interlaced 90 Hz refresh rate. You get the same line rate, equivalent to 45 Hz noninterlaced, but you (probably) get no flicker.

Some guidelines for using interlacing are as follows:

- Do not use interlacing, except maybe for comparison purposes, when your hardware can easily handle noninterlaced displays.
- Use interlacing where, in noninterlaced mode, you are too close to or over the limits of your video hardware, monitor DCF, or bandwidth, such as in the example cited earlier.
- Try to use interlacing at high resolutions so that the alternately refreshed lines are still close together.
- Do not expect miracles from interlacing. If you are exceeding your bandwidth and horizontal-scan rates (such as in the example) and

pushing the limits of your vertical-scan rates, then interlacing cannot help you. It cannot get you past the vertical-scan limits, either. Your vertical-scan rates have to be only 50 percent to 70 percent of the maximum. Even then, interlacing might not be a sure thing, but it is worth a try.

- If you are custom designing the modelines, remember to use an odd number of vertical lines.

What are the drawbacks of interlacing? If the refresh rate turns out to be too low, such that the persistence of the phosphors is too short, then the dots might noticeably lose their luminosity between electron gun passes and give you that annoying (in extreme cases, physically sickening) flicker. In addition, interlaced displays are always going to be inferior when compared with noninterlaced displays at the same vertical-scan frequency (but do not forget that they will use only half the bandwidth and half the horizontal scan rate). Anyway, flicker and the loss of quality are the most common disadvantages cited against interlacing, and those who do not understand the whys and wherefores of this technique often cite these problems.

If the phosphor dots have enough persistence and the vertical scan frequency is high enough, however, interlacing might be a good alternative in some situations—especially if you are constrained by video bandwidth, DCF, or horizontal bandwidth. Remember that television has used interlacing since its inception. That is why picture quality (resolution) has always been acceptable based on affordable components with a reasonably long, useful life.

Note also that noninterlaced line-by-line horizontal scanning was developed later than TV technology. You can probably guess, now, what high-definition TV (also called “high-density TV” or “HDTV”) is and why it is so much more expensive than conventional TV.

Just keep in mind that it is always a better idea to use noninterlaced modes at any given resolution than to use interlaced modes as long as you can establish refresh rates that are high enough to prevent flickering.

## Getting Ready for Video Configuration

---

Having gotten some basic concepts under our belts, we will gather the information we need to configure the video. The bandwidth and frequency ranges of the monitor, and the frequency of the video-adaptor’s dot clock, will determine the ultimate resolution for your display. Then, in the



upcoming exercise, we will fine-tune the video-adaptor device driver, which is something we cannot do with most software programs. As you work through the exercise, we remind you of the sections to review for concepts and definitions and to help you make various selections.

## The XF86Config File: Heart of the X Server

We constantly hear references to “the X server,” but we rarely hear a definition of it. Although many refer to individual video-driver packages as the X servers—and we do not really blame them because those video-driver and attribute files are what we focus on most—the (total) X server is actually the X11R6 software you find installed in the `/usr/X11R6` directory in conjunction with other software and configuration files. The version of X11R6 ported for use on Intel-based systems is XFree86, and that is why we refer to your particular X system as XFree86.

Thus, when XFree86 is invoked, it reads the *XF86Config* file and follows the instructions found therein. *XF86Config* is a text file found in either the `/etc/X11` or `/usr/X11R6/lib/X11` directories (one will link to the other). *XF86Config* is composed of the following sections:

- Files (file path names)
- Module (dynamic module loading)
- ServerFlags (server flags)
- Keyboard (keyboard configuration)
- Pointer (pointer configuration)
- Monitor (monitor description)
- Device (graphics device description)
- Screen (screen configuration)
- XInput (extended input devices configuration)

Without going into great detail, let us simply say that the information in these sections informs XFree86 how and when to:

- Control and route input to the correct Linux clients
- Perform basic graphic operations on one or more screens
- Control video-related attributes such as color, color depth, screen size, and so on and other attributes we discuss later in this chapter
- Allow or prohibit simultaneous access by other hosts or clients

- Load special software modules
- Control special system actions (such as core dumps and input device configurations)

We say that *XF86Config* is the heart of the X Window System because it is the most fundamental and important configuration file. Once properly configured, the *XF86Config* file is unique to the system it was configured. Unless others have absolutely identical systems (including mice, keyboards, video adapters, monitors, and so on), one system's *XF86Config* file should not be copied and passed around. In fact, doing so could be dangerous. It is always best to configure *XF86Config* for each individual system.

*XF86Config* can be configured in the following ways:

**By hand.** Constructing *XF86Config* from scratch is for seasoned veterans only. We do not recommend using this method when there are others available.

**Using Xconfigurator.** As we have already mentioned, we use this method in this chapter. It is Red Hat's GUI method and runs from the command line.

**Using xf86config.** This program is the text-mode program available with XFree86. It also runs from the command line.

**Using XF86Setup.** This works like a graphical utility. Once you are more familiar with video configuration, you might want to use this method. On the surface, it can be quite handy. If you run into trouble, however, getting out of trouble is not very straightforward.

**Using xvidtune.** This is another graphical utility that is usually employed after a basic X video configuration has been developed using one of the other utilities

For further information regarding *XF86Config*, please refer to its man pages and other information sources. Meanwhile, before we discuss Xconfigurator, we have to gather some information about the video hardware on the system for which you are going to configure X. Use the SuperProbe utility to collect this information.

## Using SuperProbe to Obtain Video Information

Before using the Xconfigurator program, we have to make sure we know the chipset and amount of video memory on the video card. It is also helpful if we know beforehand the video server we want to run.

SuperProbe can help. To use SuperProbe, though, you have to log in or su to be a root user (or, if you prefer, a superuser). Appearing next is the SuperProbe syntax and an example of what SuperProbe returns to the display:

```
[user1@hostname user1]$ su - root
Password:
*****
```

In its response, SuperProbe will present you with a warning such as the following:

```
WARNING - THIS SOFTWARE COULD HANG YOUR MACHINE.
READ THE SuperProbe.1 MANUAL PAGE BEFORE
RUNNING THIS PROGRAM.
INTERRUPT WITHIN FIVE SECONDS TO ABORT!
```

Chances are that your system will not be hung up by SuperProbe, but it is worthwhile to read its man page beforehand.

### ***SuperProbe Options***

The following are some options you can use with SuperProbe. The verbose option can be especially instructive:

- verbose.** SuperProbe will be verbose and provide lots of information as it does its work.
- no\_dac.** Skip probing for the RAMDAC type when a (S)VGA is identified. Remember that RAMDAC is yet another term for the video processors (that is, chipsets, accelerators, and so on).
- no\_mem.** Skip probing for installed video memory.
- info.** SuperProbe will print out a listing of all the video hardware that it knows how to identify.

There are other options listed in the man pages.

### ***SuperProbe Results***

SuperProbe's output is reproduced as follows. It has identified the exact chip on the video adapter. In addition, SuperProbe is suggesting that the SVGA X Server (in the first video field) is compatible for the identified chipset. We know that it is the video adapter that generates the signals to set the monitor's display. It is important that we get the correct video adapter

(or chipset) here so that all signals sent to the display (that is, to the monitor) are correct.

```
# SuperProbe
First video: Super-VGA
Chipset: Chips & Tech F65550 (Port Probed)
Memory: 2048 Kbytes
RAMDAC: Generic 8-bit pseudo-color DAC
(with 6-bit wide lookup tables (or in 6-bit
mode))
```

At this point, you could invoke Xconfigurator to begin the video setup, but first we will use it to create an *XF86Config* file.

## Using Xconfigurator to Create the Basic XF86Config File

Now that we know how SuperProbe can help us obtain some basic video information, we can focus on the main video configuration program, Xconfigurator. Xconfigurator is the program used here to create your basic *XF86Config* file. That file will be based on menu selections made after the Xconfigurator program has been invoked. As with SuperProbe, you have to be a root user (superuser) to use Xconfigurator.

The syntax for the Xconfigurator program is

```
[root@hostname /root]# Xconfigurator [-options]
```

The *XF86Config* file typically resides in */usr/X11R6/lib/X11*. A sample *XF86Config* file is supplied with XFree86. That sample file is configured for a standard VGA card and a monitor with  $640 \times 480$  resolution, which might not be optimal.

You can either take the sample *XF86Config* as a base and edit it for your configuration or let the Xconfigurator program produce a customized base *XF86Config* file for your configuration and then fine-tune it. We will not go into a detailed description of the configuration process, but you can review */usr/X11R6/lib/X11/doc/README.Config* for a detailed description of the configuration process.

For accelerated servers (including accelerated drivers in the SVGA server), there are many chipset and card-specific options and settings. The Xconfigurator program does not know about these, and on certain configurations these settings must be specified. For further details on

specific systems, refer to the server man pages and chipset-specific READMEs.

## Example Configuration

This 19-step exercise focuses on configuring your video system by modifying the *XF86Config* file by using the Xconfigurator program. This exercise might or might not be straightforward, depending on how much you learn about your video system and how well you communicate that information to the Xconfigurator program. You start using SuperProbe to get information about the video hardware. Then, armed with that information, you will invoke Xconfigurator.

Meanwhile, we comment on each Xconfigurator screen. The screens you encounter and the order in which you encounter them will depend on the choices you make along the way. If you are doing this exercise with another system at another time in another place, you might not see all of the dialog boxes shown here. We hope, but cannot guarantee, that you will not see dialog boxes that are not shown here.

We will assign just one exercise step to each screen after presenting information, comments, or warnings.

### Step 1. Invoking SuperProbe

Remember, to use SuperProbe you must log in as root. If you are already logged in as an ordinary user, use the `su` command to become a root user. Then, use the SuperProbe syntax—this time without any additional arguments or options:

```
$ su - root
Password:
*****
# SuperProbe
```

Record the following video information that SuperProbe returned to your screen:

- Video server
- Chipset
- Memory
- RAMDAC

## Step 2. Invoking Xconfigurator

As a root user, invoke Xconfigurator:

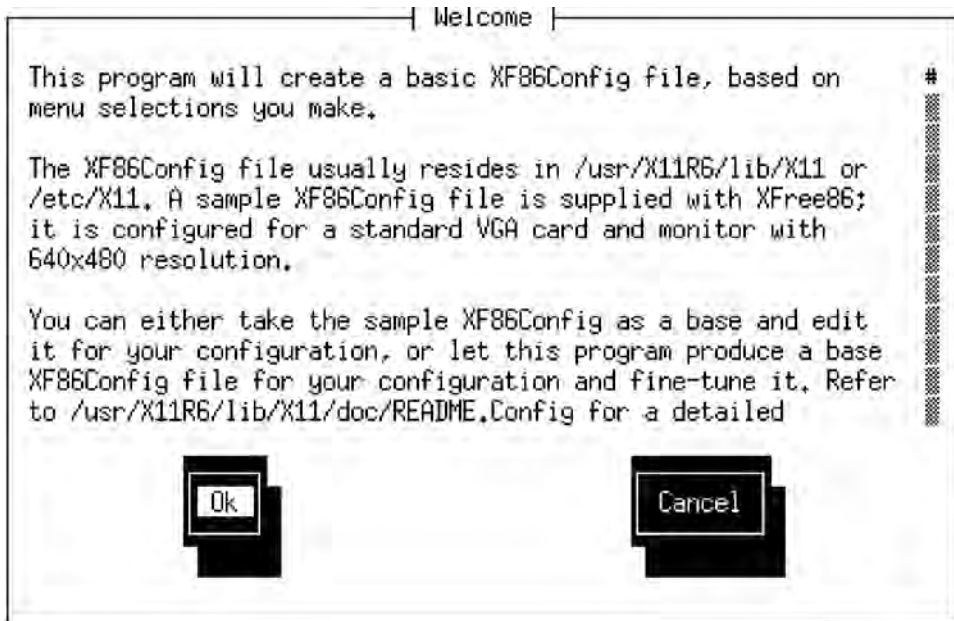
```
# Xconfigurator
```

Follow and respond to the dialog boxes, beginning with the next step.

## Step 3. Initiating Xconfiguration

This next dialog box tells you where the *XF86Config* file is located. Usually, it is found in the */etc/X11* directory; the *XF86Config* file found in the */usr/X11R6/lib/X11* directory is linked to */etc/X11*. This dialog box also warns you to know the chipset and amount of video memory.

You will likely have the name of the chipset if you have used Super-Probe. Other ways to find it involve removing the cover of your computer and examining the mainboard (if the video hardware is located there) or the video-adaptor card (if you are using one and are bypassing the mainboard). The chipset type might be printed on the chipset itself. The advice in this screen regarding accelerated servers is well worth heeding (see Figure 5.6).



**Figure 5.6** The XF86Config welcome screen.

This is one of the more useful tools that is available to help configure a fully functional *XF86Config* file. It is not available in the newest XFree86 downloads, however. For this reason, you might want to install the older version of XFree86 software.

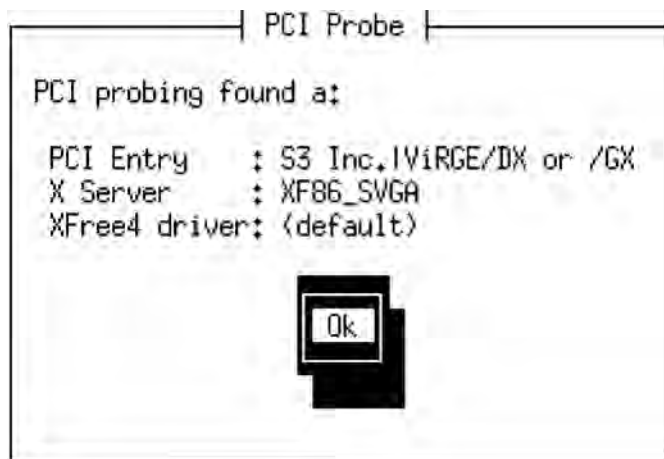
Locating the video memory also might mean that you have to remove the cover of the computer to determine whether you have a video-adaptor card or if the video hardware is on the mainboard. You might even have to check the system's BIOS (that is, enter the CMOS Setup Utility and examine the Chipset Features Setup when it boots up). Needless to say, manuals and Internet Web sites can prove invaluable in collecting this information. In the meantime, acknowledge the message and continue with the configuration by using your Tab key to select OK and then pressing Enter.

#### Step 4. Probing for Hardware

After you initiate the program, Xconfigurator will automatically probe to identify your video hardware. As you can see in Figure 5.7, it found video-adaptor hardware requiring a ViRGE/DX or ViRGE/GX driver. It also suggests using the SVGA video server.

Has the probe found something in your system? If it has, accept what it has found and continue by selecting OK and then pressing Enter to continue to the next screen.

Sometimes the probe is unsuccessful and the program asks for your help. The next section shows you the screen it presents in this event.



**Figure 5.7** PCI probe results.

### Step 5. Specifying Video-Adapter Hardware

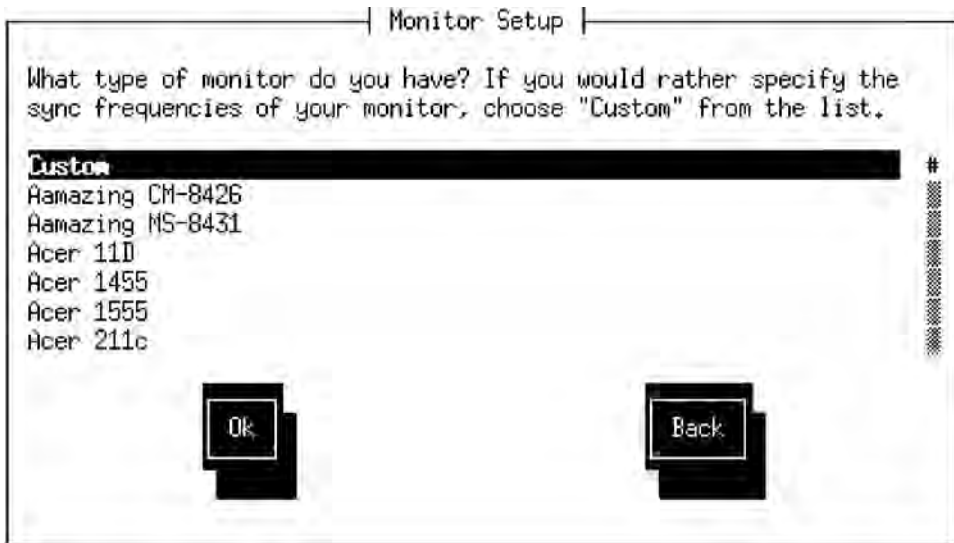
If you do not encounter another dialog box, you can move on to Step 6.

As we said earlier, Xconfigurator will automatically probe for your video hardware as soon as you initiate the program. There are times when it cannot easily determine your hardware, however. Then, it will present this dialog box. Note how it presents the name of the card in the left column and the respective driver in the right column. You can select your card (that is, your video-adapter hardware, which might or might not be a separate card). Select your video hardware, or if it is not listed, select Unlisted Card and then Tab to OK and press Enter.

### Step 6. Specifying Monitor Type

At this point, you are asked for the type of monitor attached to your system. The Linux database contains specifications for more than 200 monitors, so it is often worthwhile to scroll down through the listing presented to you. If you find your monitor listed there, you will not have to do a lot of the manual configuration. If you scroll down the list and you do not find your particular monitor, however, you must select Custom (see Figure 5.8).

Select the name of the monitor attached to your system, Tab your way to OK, and press Enter. If your monitor is not listed, select Custom and then Tab to OK and press Enter.



**Figure 5.8** Monitor setup.



### Step 7. Custom Monitor Setup

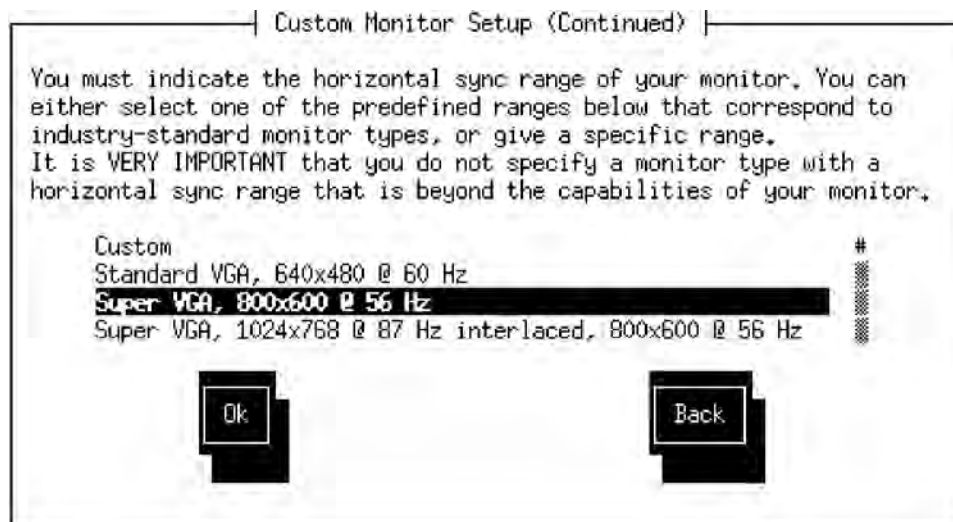
We assume that you selected Custom in the previous step. If you were able to find your monitor name in the list and selected it, then you will not see this next dialog box. You can move along to Step 10, “Setting Resolution and Color Depth.”

This screen serves as a gentle warning that the configuration might now take on a slightly more complex aspect (see Figure 5.9).

The program suggests that you check the `/usr/X11R6/lib/X11/doc/Monitors` file to see whether your monitor and its specifications are listed there. Doing that might mean exiting Xconfigurator at this point. Assume for this exercise that you checked the file and your Brand X monitor is not listed there. Well, you still have your owner’s manual.

It is very important from this point on to understand and to enter the monitor’s specifications correctly, because they represent the actual physical capabilities and not the desired capabilities of the monitor that you are trying to use.

Do you recognize the terms that Xconfigurator mentions here? You see that some are identical to names of specifications we discussed earlier in this chapter or to those that you find in your monitor’s manual, and some *resemble* those we have discussed. Recall that we mentioned that the same concepts could have several names.



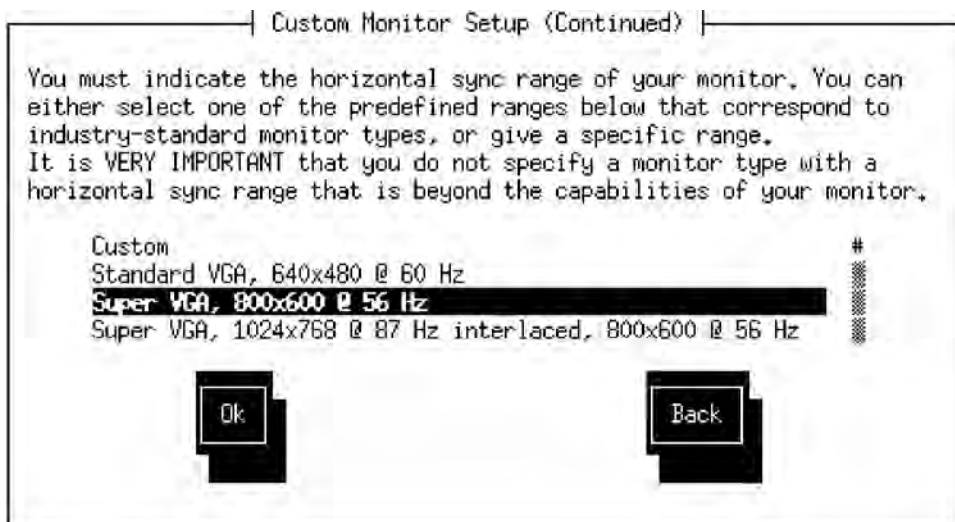
**Figure 5.9** Custom monitor setup.

Let's look at the terms in this dialog box individually. Note that you have seen "vertical refresh rate" before but that we called it simply "refresh rate." Did you remember that it is also called "vertical sync" (that is, vertical synchronization frequency)? Is the program tricky or inconsistent here? We will not venture any comment, because all of these terms are legitimate. You can see why sometimes it is not always advisable to just push ahead and start making selections and plugging in numbers, however.

Meanwhile, your response to this advisory screen is to select OK and press Enter. The message is there to remind you to be ready for the rest of the process.

### Step 8. Indicating Horizontal Sync Range

This dialog box can be slightly misleading. Figure 5.10 shows resolutions (with the industry-standard name first, followed by the corresponding line-width by screen-height specification) at certain refresh rates. Yet, the text refers to horizontal-synchronization ranges. The horizontal-synchronization rates, which we have more commonly called the line rates, are calculated for each choice by multiplying the refresh rate (that is, the Hz number) by the screen height (that is, by the second number of the resolution, such as the "480" in  $640 \times 480$ ) and adding another 4 percent to 5 percent of that number for overscan. You then compare the final calculated



**Figure 5.10** Custom monitor setup (horizontal sync).

number to the monitor's specified horizontal frequency. If it is less than the horizontal frequency, then you can be confident in choosing it.

If you do not know what the horizontal sync range is, return to Table 5.5 in the section titled "Monitor Performance and Capabilities" and consider using the resolution that corresponds to the monitor on your system. If you are not confident of that choice either, then your choice should reflect a very low line rate or a very early video standard, such as VGA or SVGA. Later, at some point when perhaps you better understand your monitor, you can reinvoke Xconfigurator and make a different choice.

For now, just select the (conservative) Super VGA standard setting, Super VGA,  $640 \times 480$  @ 56 Hz, and then Tab to OK and press Enter.

We are confident that your adapter hardware is more high-powered than this level, and we know that your monitor can handle more, too. Nevertheless, we will stick with this selection for now and make some modifications later (see Step 13, "If the Results of Xconfigurator Probe Are Not Successful") after another probe has taken place.

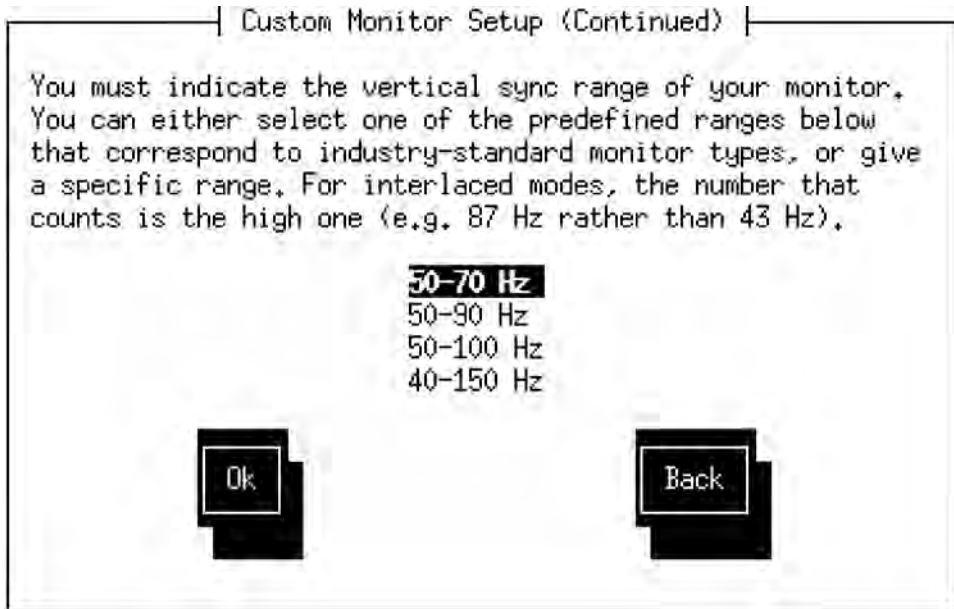
### **Step 9. Indicating Vertical Sync Range**

This dialog box, the last one in the custom setup process, addresses the vertical-synchronization range of the monitor. The listing includes several ranges found in many popular multiple-frequency monitors. Remember, the range you select should be exactly or very close to what your monitor manual lists and at any rate must exceed the refresh rate you chose in the previous (horizontal-sync range) dialog box. If there has been a mix-up, Tab to Back and press Enter. You will return to the screen in the previous step to check or change your selection there (see Figure 5.11).

Note the sentence, "For interlaced modes, the number that counts is the high one . . ." If you have selected interlaced 87 Hz in the previous screen, then the high number in the range selected here must exceed 87 Hz. Choosing 50–70 here would cause problems later. Select the appropriate vertical-synchronization range and then Tab to OK and press Enter.

### **Step 10. Setting Resolution and Color Depth**

Here, Xconfigurator will again probe the system and will, as it says, set up the default resolution and color depth. Chances are that it will come up with a very conservative default resolution setting. Do not worry. You can deal with that later.



**Figure 5.11** Custom monitor setup (vertical sync).

### Step 11. Probing with X Server Configured Thus Far

Here, the program checks with you again before it probes the video hardware and sets up default resolution and color-depth settings. The feedback will be reported in a style similar to what we show in the next step.

Press OK and press Enter to continue with the probe and setup.

### Step 12. If Results of the Xconfigurator Probe Are Successful

Assume that you know, from your manual and Table 5.5 found earlier in the chapter, that your display can handle  $800 \times 600$ . Choose Let Me Choose to do a little better than this conservative default setup, and then press Enter. This way, you will not have to edit the `/etc/X11/XF86Config` file later.

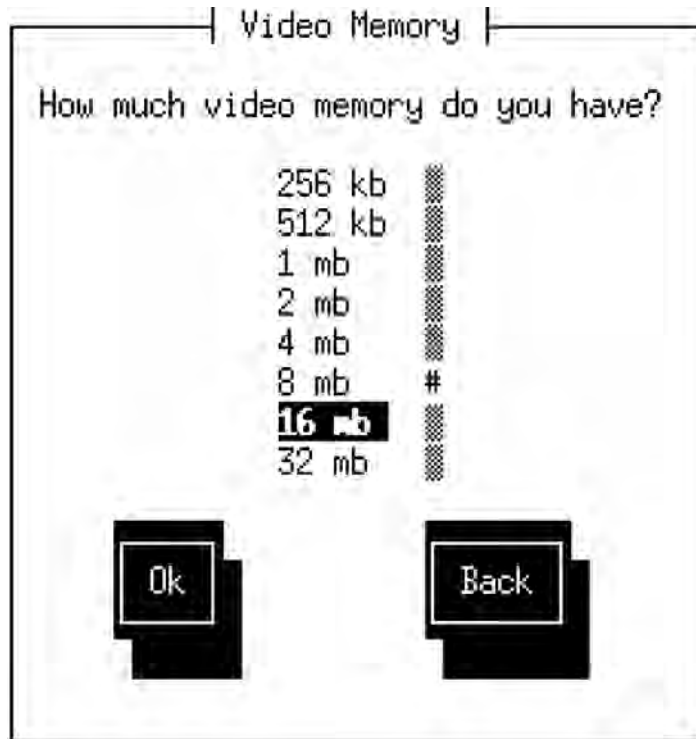
### Step 13. If Results of the Xconfigurator Probe Are Not Successful

Sometimes the probe is not successful no matter how conservative the default settings are. In this example, where Xconfigurator still cannot determine all the attributes of the video hardware (chances are the video hardware is on a board, not a card), it fails to discover how much video

memory is there. That would certainly affect the resolution settings, and then it might affect calculated DCFs and comparisons to the monitor's bandwidth.

Let's take a few steps back:

1. Press OK to take you back to the Custom Monitor Setup (Continued) screen in step 9, "Indicating Vertical Sync Range."
2. At this point, choose Back to return to the Custom Monitor Setup (Continued) screen in Step 8, "Indicating Horizontal Sync Range."
3. At that point, choose Back again to return to the initial Custom Monitor Setup screen.
4. Choose Back again. You will see a screen that resembles the following reproduction.
5. Select the amount of video memory affiliated with your video hardware (mainboard or adapter card), and then Tab to OK and press Enter. See Figure 5.12.



**Figure 5.12** Video memory.

When you back up several steps in this manner, you will be taken through a few extra dialog boxes—the ones that you would have encountered had you chosen Don't Probe at Step 10, "Setting Resolution and Color Depth."

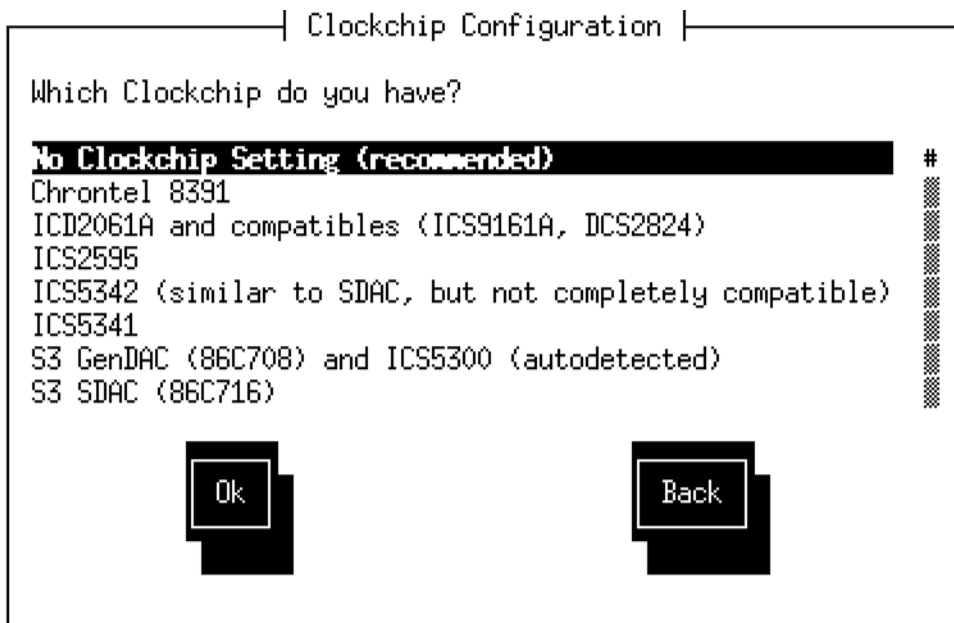
Next, you will encounter the following dialog box (see Figure 5.13).

For new technology, we agree with the program that No Clockchip Setting be selected. With older video hardware, however, especially those with video hardware on the mainboard, it might be worthwhile to go through the dozen or so choices.

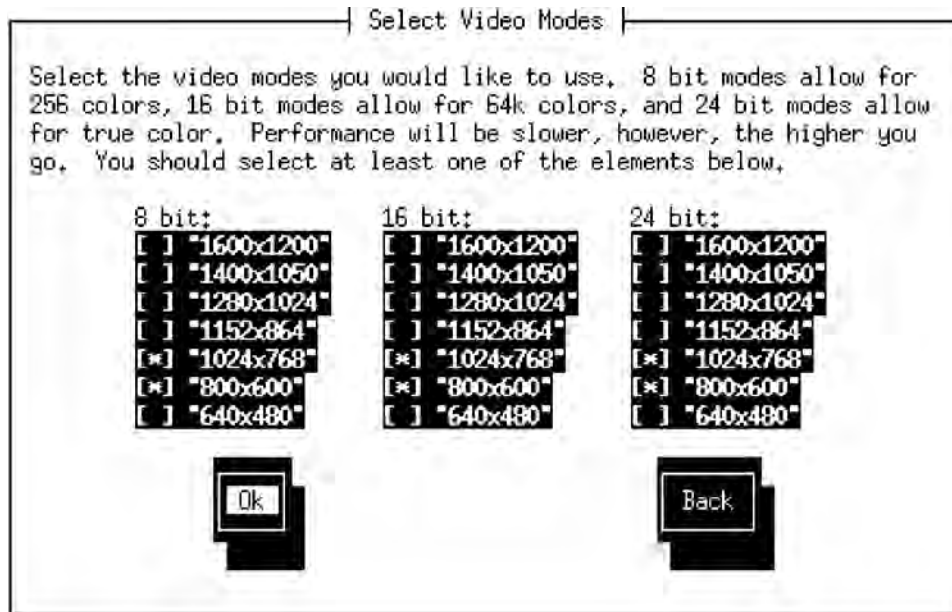
For purposes of this exercise, select No Clockchip Setting and then Tab to OK and press Enter.

#### Step 14. Selecting Video Modes

The number of video modes presented here by Xconfigurator depends on the video memory you have specified or that it has probed. Do you remember what the 8-bit, 16-bit, and 24-bit designations mean? They mean one byte per pixel, two bytes per pixel, and three bytes per pixel, respectively. The more memory, the more resolution choices Xconfigurator will present. The following screen reflects approximately 2MB of video memory. This point is also



**Figure 5.13** Clockchip configuration.



**Figure 5.14** Video modes.

where you can consult Table 5.1 in the section titled “Video Adapter Memory.” The table roughly agrees with Xconfigurator. Also see Figure 5.14.

If you have a really good monitor, you will not be able to take advantage of its capabilities unless you have adequate memory on the video card.

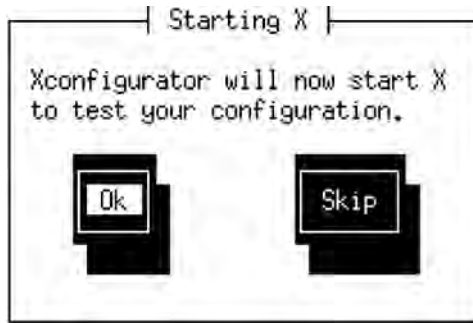
Likewise, if you have a video adapter with, say, a healthy 8MB of video memory and a monitor that is only capable of  $800 \times 600$ , you will not get the higher resolutions you want.

At this point, select the appropriate resolution(s), taking into consideration video memory, nominal monitor size, and so on. You can select more than one. The X Window System will start with the very first selection, but you can switch on the fly while in X. Later, you can modify your *XF86Config* file to comment out the resolutions you decide not to use.

Make your selections by using Tab to get to the appropriate column and then use your down arrow key. Make the selection by pressing the space bar. When all your selections have been made, Tab to OK and press Enter.

### Step 15. Starting X to Test the Chosen Configuration

If you have a good configuration, you will be very near the end of your Xconfigurator trials and tribulations. If not, then you will get another



**Figure 5.15** Starting X.

chance to modify the *XF86Config* file and reset your video parameters (see Figure 5.15).

To be able to select OK or Skip at this stage is convenient. If you are not sure you have configured your video correctly, then you can at least finish the basic Linux installation and plan to reconfigure your video (to use the X Window System and its assorted window managers) later. Be confident and select OK, however, and then press Enter to see whether you were successful.

#### **Step 16. Can You See This Message?**

If the text is in the middle of a cyan- or teal-colored display, then you can continue by selecting Yes. If the text appears in a ridiculous location on the display or at the bottom of the screen, then you should probably select No and pass through the configuration again. If you do nothing, or if you cannot see the message, the program will time out shortly and automatically return you to the configuration process.

Let's assume that the box is nicely placed in the middle of the cyan-colored screen. Select Yes by using your mouse, and you will be presented with the next dialog box.

#### **Step 17. Do You Want to Automatically Boot into X?**

At this point, we suggest that you select No. If you select Yes and all goes well, then certainly Linux will start the X Window System immediately and will automatically boot into the X Window System every time the system is booted. If you select Yes and all does *not* go well, then you might be stuck with a frozen screen in a bad configuration, requiring you to take remedial action or worse.

Thus, until you are satisfied with your video configuration, click No at this juncture. You can always configure X to start at bootup at some later date.



**Step 18. Reviewing the Configuration File Prior to Running X**

This display might or might not be the look of success. It is a good sign, however, that you are getting closer. The Xconfigurator program suggests here that you take a good look at the *XF86Config* before running *startx*, which is a very good idea. If the newly configured file works at all at this point, consider making a copy of it and storing it somewhere for safekeeping. The instructions regarding cycling through the various modes are only valid if you previously chose more than one valid video mode.

The program also suggests that you exit the X Server by using the Ctrl-Alt-Backspace keys. This method is rather brutal and should be avoided, however. There are other proper logout procedures available from within X. If something goes wrong while in X, however, that essential sequence can be necessary.

Finally, the program suggests you refer to the *README.Config* file. While not a bad idea, the text in that file pales in comparison to the information that can be found at the Internet site <http://metalab.unc.edu/LDP/HOWTO/XFree86-video-Timings-HOWTO.html>.

At this point, click OK and you will be transported out of Xconfigurator and back to the Linux command-line prompt. You are now free to check your handiwork by typing *startx* at the command prompt and pressing Enter. Once you are in X, you can cycle through your video modes by using the Ctrl-Alt-+ or Ctrl-Alt- key sequences.

At some point in your configuration process, you might be presented with the dialog box discussed in the next step. We certainly hope not, but let's have a look at it anyway.

**Step 19. Problems with Chosen Configuration**

It is always a disappointment to encounter this next dialog box. You do not usually encounter it at the end of the Xconfigurator process; instead, it often happens several steps earlier.

This message usually means that your chosen video modes disagree with your video memory specification, but it can also mean several other things. Your best bet is to go back and try again, because sometimes the problem might only be an oversight in a recent dialog box. Sometimes, however, the best strategy is to simply quit and restart, perhaps after checking your hardware via SuperProbe or another utility.

There are known problems with Xconfigurator, however. The problem involves the use of the *-expert* option. What happens is that the video card that is probed for when Xconfigurator initializes is the one entered in */etc/X11/*

*XF86Config*, despite ignoring the specific card that you might choose. It appears that the configuration utility cannot shake the name of the previous card used. The settings for the manually selected card are still used in *XF86Config*. The information it is accessing is the `Device="Cardname"` string in the *XF86Config* file. In other words, you have to make a note of the card that you need and manually edit the *XF86Config* file to reflect the correct card and chipset. Bugzilla is currently working on submission number 18845 with regard to this issue.

## How to Install X from Scratch

---

As we previously mentioned in the installation chapter, when you choose to install a server class, system X is not included as one of the base rpm sets. If you did not remember to add it to the list of additional packages, then you will have to install all the base X rpms along with the accompanying desktop files from the beginning. The files here were sourced from the [www.redhat.com](http://www.redhat.com) site rather than from [www.xfree86.org](http://www.xfree86.org) simply because the correct recommended level of X packages were nicely organized underneath the respective distribution levels.

### Step 1. Install the X Packages

```
rpm -Uvh --nodeps X11R6-contrib*
rpm -Uvh --nodeps XFree86-3.3*
rpm -Uvh --nodeps XFree86-libs*
rpm -Uvh --nodeps Xconfigurator*
rpm -Uvh --nodeps fvwm*
rpm -Uvh --nodeps xinitrc*
rpm -Uvh --nodeps gtk+-*
rpm -Uvh --nodeps tcl-*
rpm -Uvh --nodeps tk-*
rpm -Uvh --nodeps SuperProbe*
```

### Step 2. Determine the Required X Server

If you do not know what the video details are, then run both SuperProbe and Xconfigurator to see what RPM it says is not installed. At this point, you will be able to install that by hand (here is an example).

```
rpm -Uvh XFree86-SVGA*
```

### Step 3. Run Xconfigurator

Run the Xconfigurator to configure X on your system. At this point, you can startx. This configuration is bare bones X without a desktop. If you would like to have a desktop, you now have to install those packages, as well.

If the Xconfigurator throws an error about not having a base X server to use, try installing the following servers for the Xconfigurator.

Here are the drivers for XFree86 3.3.6 for various video card chipsets, one of which might be needed by Xconfigurator:

```
XFree86-3DLabs-3.3.6-33.i386.rpm  
XFree86-8514-3.3.6-33.i386.rpm  
XFree86-AGX-3.3.6-33.i386.rpm  
XFree86-FBDev-3.3.6-33.i386.rpm  
XFree86-I128-3.3.6-33.i386.rpm  
XFree86-Mach32-3.3.6-33.i386.rpm  
XFree86-Mach64-3.3.6-33.i386.rpm  
XFree86-Mach8-3.3.6-33.i386.rpm  
XFree86-Mono-3.3.6-33.i386.rpm  
XFree86-P9000-3.3.6-33.i386.rpm  
XFree86-S3-3.3.6-33.i386.rpm  
XFree86-S3V-3.3.6-33.i386.rpm  
XFree86-SVGA-3.3.6-33.i386.rpm  
XFree86-VGA16-3.3.6-33.i386.rpm  
XFree86-W32-3.3.6-33.i386.rpm
```

### Step 4. Install the Desktop

The list of packages to install is very large—unmanageably large. Lists of the packages you require are listed next. Unless you are Barbara Blackburn, you are not going to want to type these entire rpm package files. On the distribution, there is a file called *comps* in the *RedHat/RPMS/base* directory that can be used as a source for the package names.

The definitive list of packages you require will be listed under the section headings: X Window System, KDE, and GNOME. You can copy the packages out of this file and use an editor to prefix the names with `rpm -Uvh -nodeps -force` and append an asterisk (\*) to the end of the name to expand it. This process will do an upgrade/install with all the packages listed in your new file.

There is one alternative if you have a set of CDs and you are not afraid to reconfigure your system. You can initiate an upgrade and ask to have the entire X Window and Desktop packages included. Just make sure that you do not format any partitions during the process.

For those who do not have a set of CDs, here is the X Window System and Printer group list file from */RedHat/base/comps* that is required:

chkfontpath  
groff-perl  
perl  
tcsh  
LPRng  
krb5-libs  
mpage  
rhs-printfilters  
libpng  
ghostscript  
freetype  
pnm2ppa  
ghostscript-fonts  
urw-fonts  
XFree86-xfs  
XFree86-libs  
usermode  
gtk+  
libjpeg  
libtiff  
perl  
netpbm-progs  
ImageMagick  
imlib  
libungif  
ORBit  
audiofile  
gnome-audio  
gnome-libs  
esound  
libpng  
XFree86-tools  
XFree86-twm  
XFree86-xdm  
xinitrc  
XFree86  
XFree86-75dpi-fonts  
freetype  
XFree86-libs  
Mesa  
Xaw3d  
xsri  
Xconfigurator  
control-panel  
expect  
fortune-mod  
gv  
indexhtml  
netpbm  
perl

rxvt  
m4  
mkxauth  
sharutils  
modemtool  
netcfg  
printtool  
tkinter  
pygnome  
pygtk  
python  
pythonlib  
timetool  
tcl  
tclx  
tix  
tetex-xdvi  
tetex-fonts  
tk  
tksysv  
rpm-python  
python-xmlrpc  
pygtk-libglade  
libglade  
libxml  
xloadimage  
xscreensaver  
audiofile  
bug-buddy  
control-center  
desktop-backgrounds  
dia  
ee  
esound  
extace  
gdk-pixbuf  
gdm  
glade  
glib  
gtk+  
gtk-engines  
mc  
gftp  
gedit  
gmc  
gnome-audio  
gnome-audio-extra  
gnome-applets  
gnome-core  
gnome-libs

gnome-pim  
gnome-users-guide  
gnome-utils  
gnorpm  
gnotepad+  
gnumeric  
gqview  
gtop  
guile  
umb-scheme  
imlib  
imlib-cfgeditor  
libglade  
libghttp  
libgtop  
librep  
libxml  
magicdev  
ORBit  
pygnome-libglade  
pygtk-libglade  
sawfish  
rep-gtk  
rep-gtk-gnome  
rep-gtk-libglade  
xchat  
xscreensaver

Good luck.



# Configuring Printers

In this chapter, we will illustrate how to configure both a local and a remote printer. The example printer is a Hewlett-Packard LaserJet 6 in both scenarios. In the local printer scenario, the printer is attached directly to the computer with a parallel cable. In the remote scenario, HostC will use the printer on HostA over the network.

The following instructions include the use of `printtool`, a printer configuration utility with a GUI. Consequently, it is assumed that you have worked through the video configuration chapter (Chapter 5) and have a GUI available. Unfortunately, this tool is almost the only one available now with this distribution. There are, of course, tools within the actual GNOME and KDE desktops that you can also use. If you understand this tool, you will be able to handle any desktop tool presented.

## Basic Commands and Utilities

---

The three main print mechanisms are `lpr` (line printer command), `lpd` (line printer spooler daemon), and `printcap` (printer capability database). `lpr` uses `lpd` to print files. When `lpd` receives a request to print a



file, it checks the *printcap* file to determine whether the printer requested is local or remote. If the request is for a local printer, *lpd* checks for the queue—and if the print queue is up, it sends the print job to the printer via the queue. If the request is for a remote printer, *lpd* opens a connection to the remote system and transfers both the control file and data file to the queue on the remote system. Before any printing services will work, the *lpd* services must be installed on the system.

You can ensure that all the components for *lpd* are in place by querying Linux for the necessary package. In older distributions, this package was just *lpr-0.50-4*. The new package is called *LPRng-3.7.4-22*. These instructions will work for either.

```
[root@HostA lpd]# rpm -aq | grep -i lpr
LPRng-3.7.4-22
```

To find out more about this new *LPRng* package, use *rpm* with the *-qip* options:

```
[root@HostA RPMS]# rpm -qip LPRng-3.7.4-22.i386.rpm | more
Name       : LPRng           Relocations: (not relocateable)
Version    : 3.7.4           Vendor: Red Hat, Inc.
Release    : 22              Build Date: Thu 29 Mar 2001
Install date: (not installed)
Build Host: porky.devel.redhat.com
Group      : System Environment/Daemons
Source RPM: LPRng-3.7.4-22.src.rpm
Size       : 4837549
License: GPL and Artistic
Packager   : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL        : http://www.lprng.com
Summary    : LPRng Print Spooler
Description :
```

The *LPRng* software is an enhanced, extended, and portable implementation of the Berkeley *LPR* print spooler functionality. While providing the same interface and meeting RFC1179 requirements, the implementation is completely new and provides support for the following features: lightweight (no databases needed) *lpr*, *lpc*, and *lprm* programs; dynamic redirection of print queues; automatic job holding; highly verbose diagnostics; multiple printers serving a single queue; client programs do not need to run SUID root; greatly enhanced security checks; and a greatly improved permission and authorization mechanism.

They have only renamed the package from `lpr` to `LPRng`. Both contain the same utilities. When the `lpd` daemon is active, you can still use the `lpc`, `lpq`, and `lpr` commands to manage printing.

- `lpc` is the line printer control program. Among other things, you can use `lpc` to enable or disable a print queue, get the status of a print queue, and restart a daemon should one daemon die (leaving print jobs unprocessed in a queue). Check the `lpc` man pages for the full capabilities of this utility.
- `lpq` is the spool queue examination program. Use it to display and manipulate jobs in a print queue by using the `lpq -l -Pprinter` command. To remove a print job from the queue, use the `lpq -Pprinter job_ID` command.
- The `lpr` command is a companion to the `lpd` daemon, and the daemon must be available for the two to work together. `lpr` is used to submit jobs to the printer and has many options.

It is possible to send a job directly to the printer without using `lpr`, but this action is not recommended in a shared printing environment. Print jobs will collide when multiple users send print jobs to the same printer at the same time. The `lpr` command solves this issue by sending signals to the `lpd` daemon. All `lpr` processes communicate with a single `lpd` daemon that organizes the requests. When a job is printed with `lpr`, `lpr` assembles the data and copies it to the spooling queue. At the same time, `lpr` adds to the assembly any instructions passed as arguments to the `lpr` command, which are set by using environment variables and system print defaults.

In addition, the `lpr` command uses the `/etc/printcap` file to determine the printer geometry (that is, the printer-to-print queue relationship) and additional instructions that might be in the `/etc/printcap` file, such as remote printing sets.

You can ensure that `lpr` is available by querying for the `LPRng` package, as seen in the following example. If the package is installed, then `lpr` should be available.

```
[root@HostA /root]# rpm -q LPRng
LPRng-3.7.4-22
[root@HostA /root]#
```

When executed, the `lpr` command will create two files in the spool directory: a control file with the name `cfid_number`, which contains information

about the print job and the owner's name, and a data file, *dfid\_number*, which is the file to be printed.

The */etc/printcap* file is the configuration file for *lpd*. If you use the *printtool* utility, you will not have to configure any of these services manually, including the */etc/printcap* file, but it is a good idea to know what this file does and how to interpret it. A typical *printcap* file is reproduced as follows. This *printcap* file is the result of the next scenario using the *printtool* utility.

```
[root@HostA RPMS]# cat /etc/printcap
# /etc/printcap
#
# DO NOT EDIT! MANUAL CHANGES WILL BE LOST!
# This file is autogenerated by printconf-backend during lpd init.
#
# Hand edited changes can be put in /etc/printcap.local, and will be
included.

HostA_q|draft_q:\
    :sh:\
    :ml=0:\
    :mx=0:\
    :sd=/var/spool/lpd/HostA_q:\
    :lp=/dev/lp0:\
    :lpd_bounce=true:\
    :if=/usr/share/printconf/mf_wrapper:

#####
##Everything below here is included verbatim from /etc/printcap.local ##
#####
# printcap.local
#
# This file is included by printconf's generated printcap,
# and can be used to specify custom hand edited printers.
```

Each entry in the */etc/printcap* file represents a configuration step that we are going to perform next. This file is constructed for you by the *printtool* utility. You should be able to interpret this file once you have worked through the next step-by-step example given to create local printing.

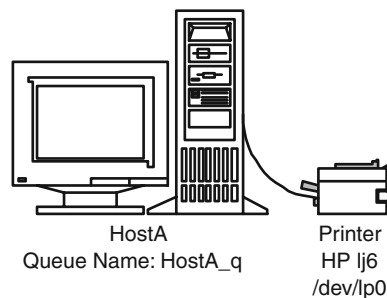
There are two things about the */etc/printcap* file that you must keep in mind. First, if you make changes to this file either directly or by using the *edit* function in the *printtool* utility, you have to refresh the *lpd* daemon. The *printtool* utility updates the */etc/printcap.local* file with new entries, and the entries are not moved over to the */etc/printcap* file until the

lpd daemon is restarted. Second, these are a team of files that work together to assemble remote printing. For example, there must be a corresponding entry in either the *hosts.lpd* or the *hosts.equiv* file(s) of the remote system for remote printers to gain access to status information. The lpd daemon will reject a print request from any remote system that is not listed as a trusted host in one of these files. We discuss these files in more detail later in the chapter. The premise here is that if the remote printer is unavailable, the job will remain in the local queue until it receives a signal that the remote queue is available. At this time, the job will be transferred across the network to the other queue. In addition, the jobs are not removed from the local queue until a signal is received from the remote system that the job is in the remote queue.

## Configuring Local Printing

Before we go through the step-by-step example procedure, we would like to briefly discuss the larger scope. It is our experience that most try to make this situation more complex than it should be. Do not fall into that rut. There are overwhelming sets of configurations. Most of these you will never have to deal with, but collectively everyone will use the parameters at one time or another. Because there are thousands of printers on the market, however, some with correct drivers and some without, there is an undetermined list of ways to configure a printing scenario. We are going to show you a very straightforward example, starting with local printing then moving on to remote printing (see Figure 6.1).

The trick to creating a simple print environment is to create a print queue first, and during the printq definition process, the printer will be chosen. In other words, you create a queue first and associate the printer



**Figure 6.1** Local printing configuration.

with it afterwards. This point is where the trouble begins. Do not try to create the printer first. As odd as it sounds, Unix printing is almost backwards to most other printing environments whereby the queue looks like the printer to the user who is using it. The reason is because in Unix, the print queue and the print driver are synonymous. In normal PC environments, the print stream is processed prior to entering the print queue. In such a case, the print queue is a passive entity. In Unix, the print stream is formatted when it hits the queue. The print queue is active. For this reason, as soon as you create a print queue, it wants to know immediately what printer this queue is going to service so that it can format the print job correctly. Therefore, we start with defining a print queue and then work out the details of the printer and the printer driver later.

### **Step 1. Define the Queue**

To configure a local printer, begin by logging in as a regular user and then switching to root user. Enter `printtool` at the command line.

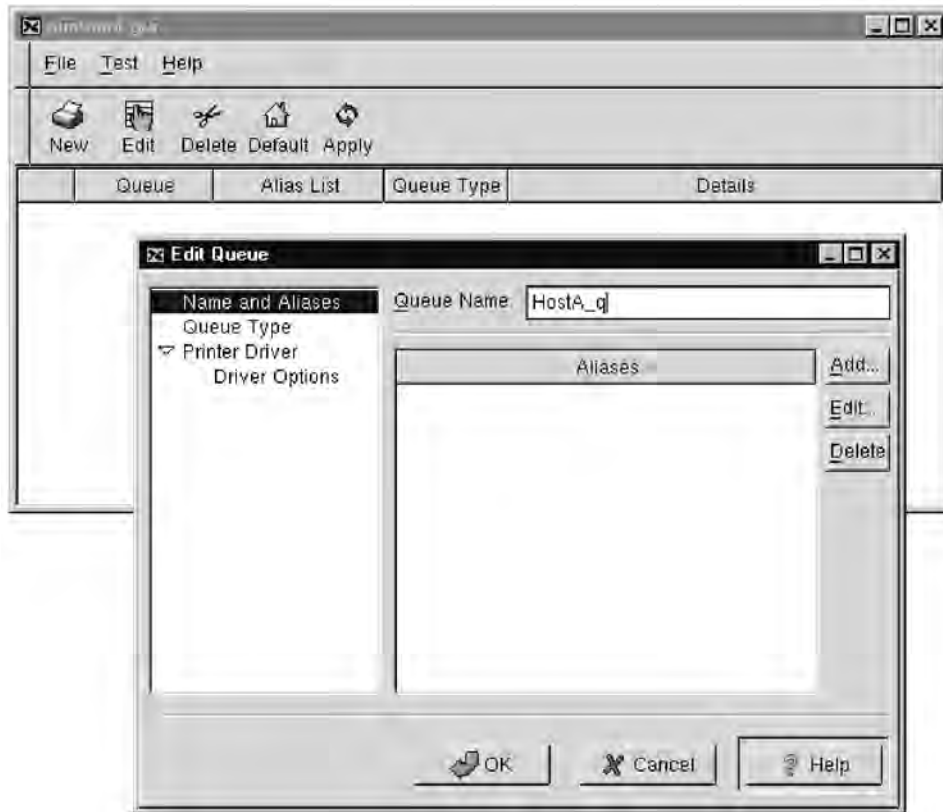
When the first `printtool` panel appears, enter an appropriate name in the Queue Name dialog box. We are going to use `HostA_q` so that we will not confuse our users after remote printing is set up. See Figure 6.2.

```
[root@HostA ~]# printtool
```

Enter an Alias name for the existing Queue Name if you wish. This name can be used to help users and applications access the services by an alternative name. For example, someday you might wish to change the system in which this print queue resides. If all the applications on the network have been configured to use the alias name and not the real name, then your job is easy. All you have to do is give the new printer the same alias reference as the last printer, and now you do not have to reconfigure all those network applications (they are all now using the new printer). See Figure 6.3.

### **Step 2. Define the Queue Type**

Now that the queue is made, the `printtool` immediately wants to know where the printer is going to be attached. In our first example, it will be a local printer. The printer device is `/dev/lp0`, which represents the local parallel port on the same system on which the print queue is being created. This point is a very important, because once local printer port `/dev/lp0` has been indicated, it will not attempt to send any information to the network. Notice also that this action is labeled Queue Type and not printer. See Figure 6.4.



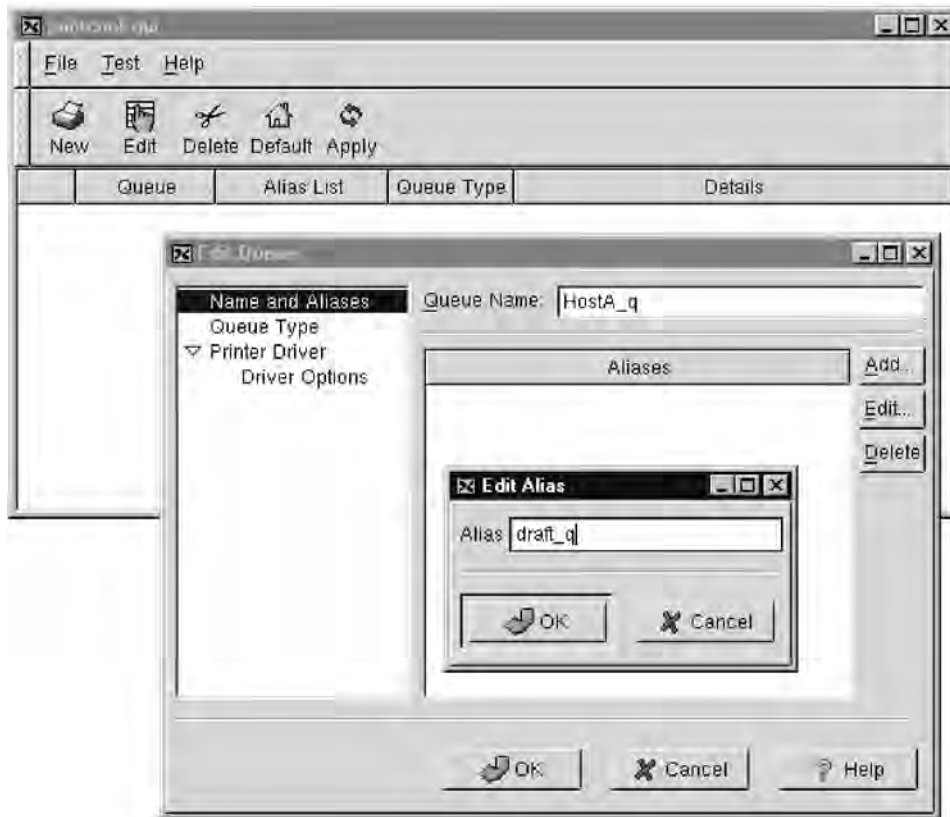
**Figure 6.2** Specifying the queue name.

### Step 3. Select the Printer Driver

The next step is to indicate the printer driver. This instance is when the printer is created. When you chose the Printer Driver option, a list of printer manufacturers will appear. Navigate to the manufacturer or your printer and select your printer. This printer driver is the one that you are selecting, and it should be accurate. In our case, there are two drivers available for the LaserJet 6L: `lj5gray` and `ljet4`. Keep copious notes here, because if you are going to set up remote printing from other systems, they should be configured with the same parameters (see Figure 6.5).

### Step 4. Customize the Driver Options

Driver options are specific for the printer that you have chosen. If you do not see the appropriate driver options that you expect, you probably have

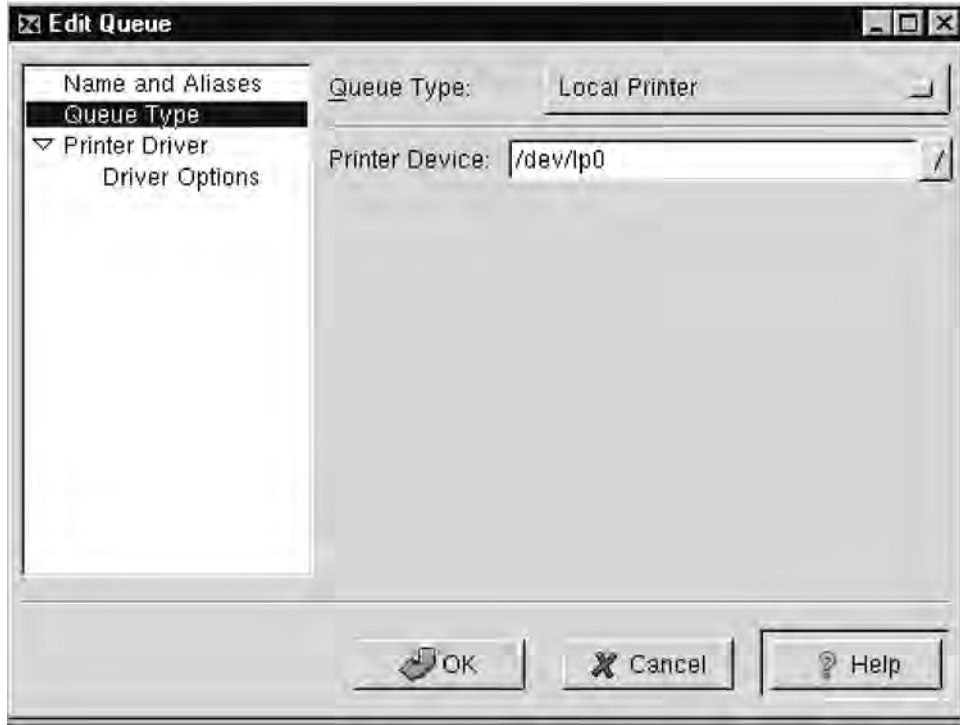


**Figure 6.3** Assigning an alias for the queue name.

the wrong driver or a different driver. For example, if you expected to see all kinds of options on envelope trays and you do not, then you have the wrong printer selected. These printer options are a function of the printer and will change from printer to printer (see Figure 6.6).

### Step 5. Enable lpd

When you are finished with the printer options, the new Queue/Printer entry will appear in the printtool. You can revisit this entry at any time to reconfigure by selecting the entry and choosing Edit. The printing services will not work yet. At this point, you have to start and or refresh the lpd daemon before the new queue and printer are enabled. Just make sure that whenever you add a new queue or change an existing queue, you must refresh the lpd. There is a quick Restart lpd option in the File drop-down menu. See Figures 6.7 and 6.8 for more information.



**Figure 6.4** Specifying the queue type.

### Step 6. Test the Printer

The first thing that you should try to do at this point is test the printer. You have the Print ASCII Test Page option under the Test drop-down menu (see Figure 6.9).

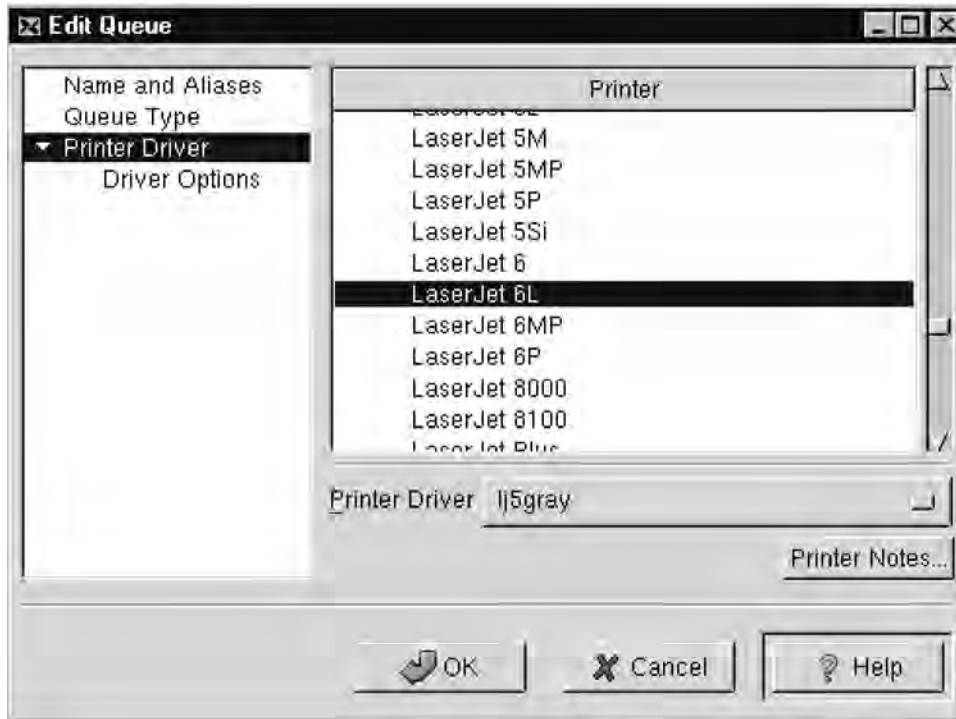
At the end, you should have successfully created a simple local printing scenario.

### Step 7. Test Printing Commands

At this point, you should be able to use, manipulate, manage, and control this print queue/printer by either using the `printtool` or the `lpc`, `lpq`, and `lpr` command-line utilities.

`lpq` here shows that there are no jobs currently in this queue. Using the `lpc` command, we have put a holdall on the queue so that the jobs do not flow through faster than we can see them. Also, because we only have the one queue on the system, we do not have to use the queue name as an argument to all of the `lpc`, `lpq`, and `lpr` commands. If you have more than one queue, then you would have to, however.



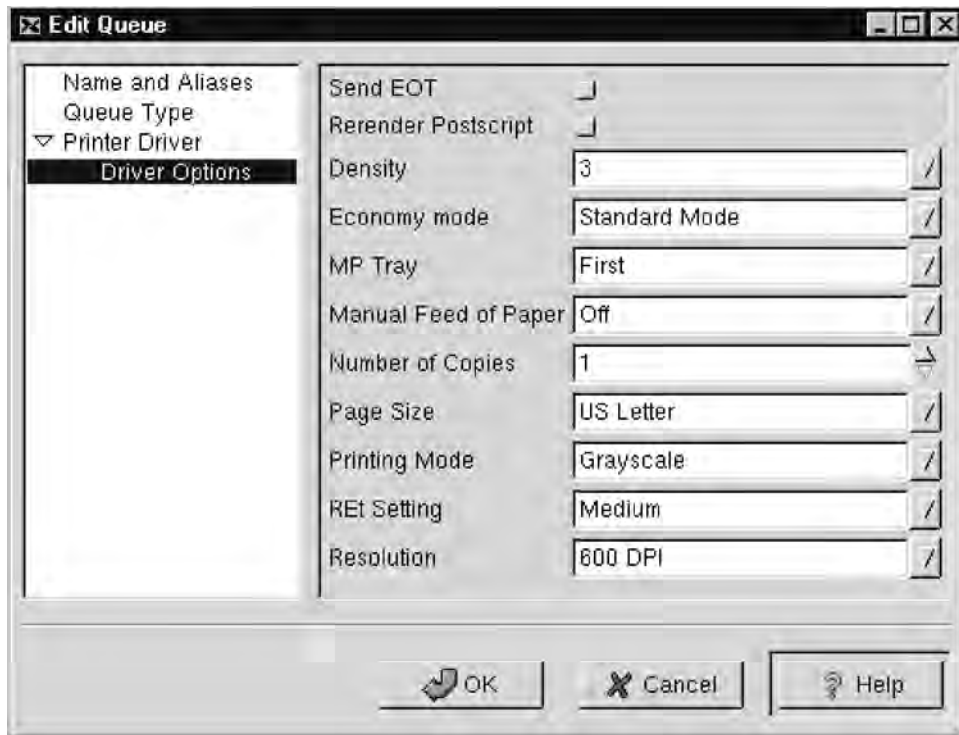


**Figure 6.5** Printer driver.

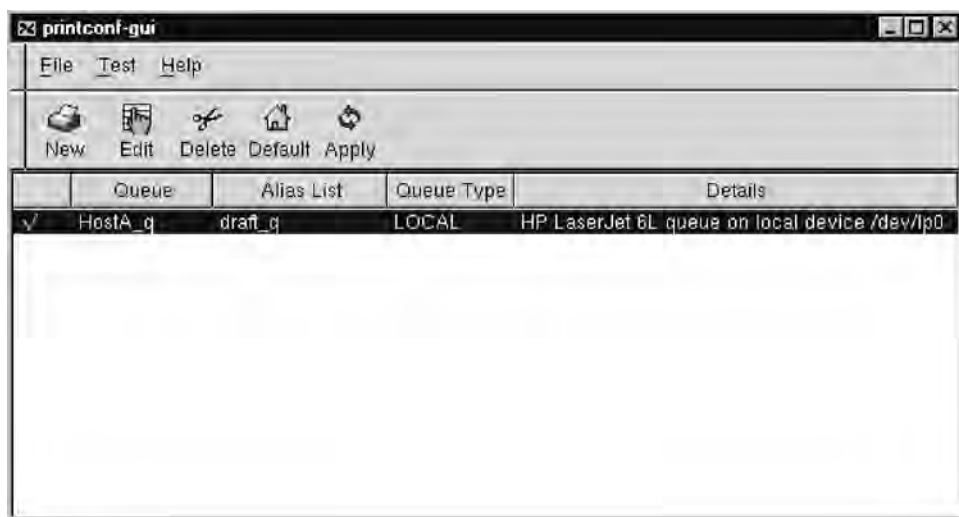
```
[root@HostA /root]# lpc holdall
[root@HostA /root]# lpq
Printer: HostA_q@localhost 'draft_q' (holdall)
Queue: no printable jobs in queue
Status: job 'cfA828localhost.localdomain' removed at 16:37:05.490
```

Send a print file called *memo.txt* to the queue by using `lpr`, and view the queue entry with `lpq`:

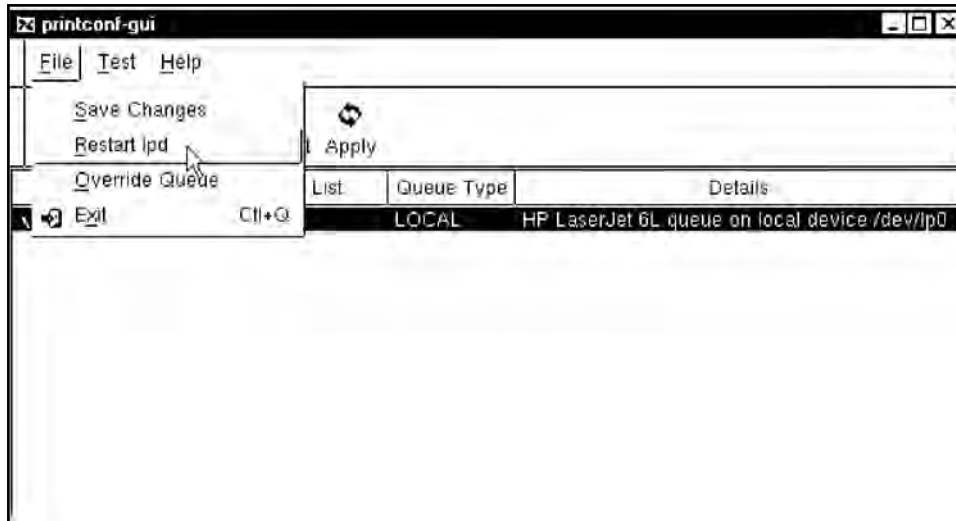
```
[root@HostA /root]# lpr memo.txt
[root@HostA /root]# lpq
Printer: HostA_q@localhost 'draft_q' (holdall)
Queue: no printable jobs in queue
Holding: 1 held jobs in queue
Server: no server active
Status: job 'cfA828localhost.localdomain' removed at 16:37:05.490
Rank  Owner/ID                Class Job Files      Size  Time
hold  root@localhost+849        A    849 memo.txt        53    16:38:09
```



**Figure 6.6** Additional driver options.



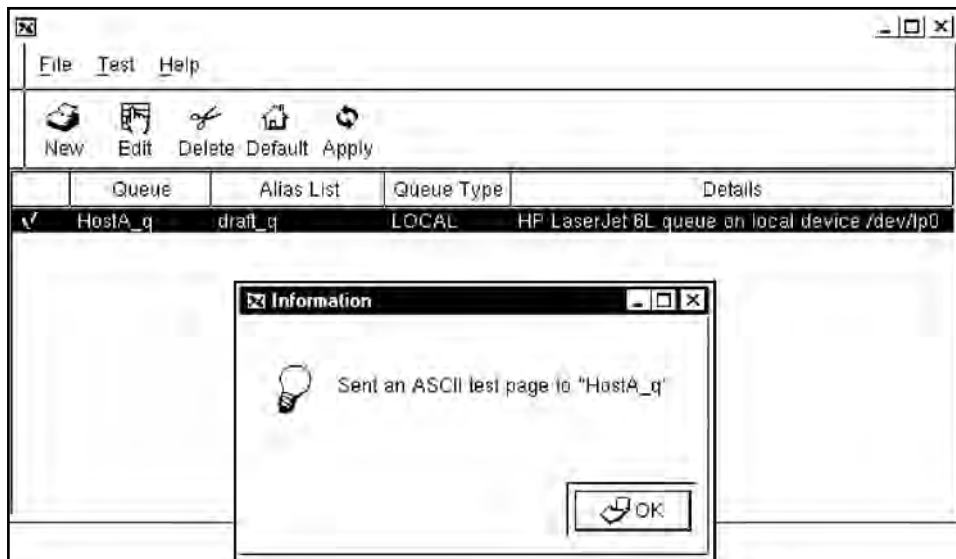
**Figure 6.7** A check mark denotes the default printer.



**Figure 6.8** Start or restart the lpd daemon.

Send another print file called `/etc/hosts` to the queue by using `lpr`, and view the queue entry with `lpq`:

```
[root@HostA /root]# lpr /etc/hosts
[root@HostA /root]# lpq
```



**Figure 6.9** Print a test page.

```
Printer: HostA_q@localhost 'draft_q' (holdall)
Queue: no printable jobs in queue
Holding: 2 held jobs in queue
Server: no server active
Status: job 'cfA828localhost.localdomain' removed at 16:37:05.490
Rank   Owner/ID                      Class Job Files      Size Time
hold   root@localhost+849              A    849 memo.txt       53 16:38:09
hold   root@localhost+853              A    853 /etc/hosts       152 16:38:33
```

Remove job number 853 from the queue by using the `lprm` command, and view the queue entries:

```
[root@HostA /root]# lprm 853
Printer HostA_q@localhost:
  checking perms 'root@localhost+853'
  dequeued 'root@localhost+853'
[root@HostA /root]# lpq
Printer: HostA_q@localhost 'draft_q' (holdall)
Queue: no printable jobs in queue
Holding: 1 held jobs in queue
Server: no server active
Status: job 'cfA828localhost.localdomain' removed at 16:37:05.490
Rank   Owner/ID                      Class Job Files      Size Time
hold   root@localhost+849              A    849 memo.txt       53 16:38:09
```

Release all the print jobs in the queue by using the `lpc` command to print and view the `printq` entries:

```
[root@HostA /root]# lpc release all
Printer: HostA_q@localhost
HostA_q: selected 'root@localhost+849'
kill server PID 0 with User defined signal 1
HostA_q@localhost.localdomain: started
[root@HostA /root]# lpq
Printer: HostA_q@localhost 'draft_q' (holdall)
Queue: no printable jobs in queue
Status: job 'cfA849localhost.localdomain' removed at 16:38:53.338
[root@HostA /root]#
```

## Technical Printing Details

The Spool Directory is the `/var/spool/lpd/HostA_q` directory. By default, and by convention, it reflects the name of the print queue. You are allowed only one print queue per spool directory by virtue of assigning alias queue names. Do not confuse the fact that you have multiple names for one print

queue, with the print queue representing one physical location on the hard disk called `/var/spool/lpd/qname`. We recommend that you retain the spool directory name, `/var/spool/lpd/HostA-q`; it is proper form to keep the name of the spool directory the same as the name of the queue.

Each spool directory contains four files—*eq*, *errs*, *lock*, and *status*—with `rw-rw-r-` permissions. These permissions are important and would cause the print process to fail if they were changed. Utilities such as `printtool`, `lpc`, and `lprm` will take care of this task for you, but if you are restoring a system from a backup or are manually attempting to create a print scenario, this situation could become an issue. For example, we would recommend that you do not delete the files in this directory to remove print jobs. Use the `lprm` command instead.

The `lpd` daemon should be configured to start at system boot. This behavior is the default after installation, but you should check it anyway. During the `lpd` startup process, the `/etc/printcap` file is read to identify the sections that apply to any printer attached to the system. The `lpd` daemon also applies any input filter that has been applied in the configuration. The only time you should need to refresh the `lpd` daemon is if you change any parameters in your configuration. You can use the `printtool` or refresh the daemons directly by using the `/etc/rc3.d/S60lpd` scripts.

## Configuring Remote Printing

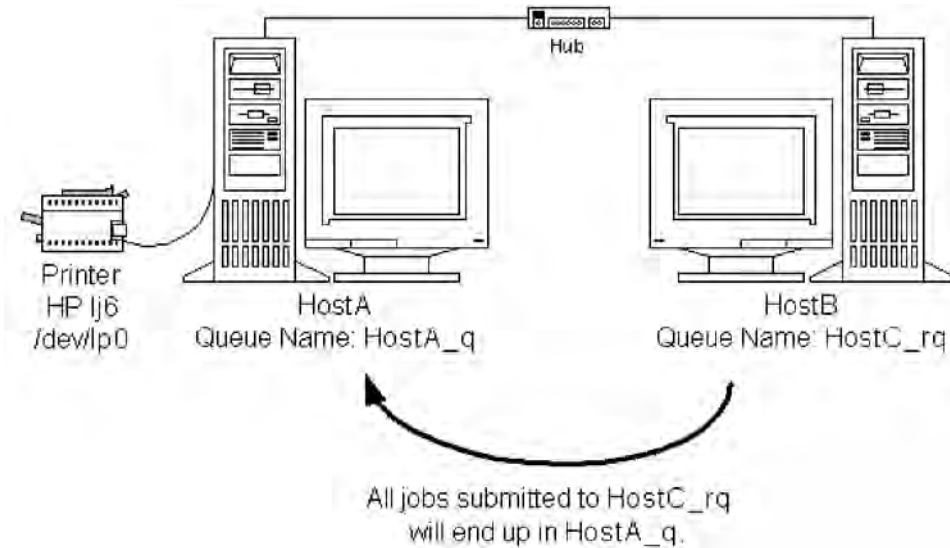
---

In this section, we assume that you have successfully worked through the previous example and you have local printing working. For our configuration, this local printer is `HostA`. We plan to have `HostC` use the printer on `HostC`. Therefore, we will configure `HostC` for remote printing (see Figure 6.10).

Setting up local printing is a prerequisite to remote printing, because the latter simply sends a print request across a network to be printed on the already-functioning system.

### Prerequisites

The basis for remote Unix printing is a workable TCP/IP configuration. Before you can address remote printing, you should test and confirm the TCP/IP communication between the two systems. Make sure that you can ping from one system to the next. You also have to confirm that the name resolution is working, which means you should be able to ping success-



**Figure 6.10** Remote printing configuration.

fully by using the host names. In the example here, we are using flat TCP/IP host files:

### Step 1. Confirm the TCP/IP Settings

The `/etc/hosts` file on HostA has been updated to include HostC, and a ping test was performed:

```
[root@HostA /root]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1    localhost.localdomain  localhost
192.168.1.206 HostA          #This system is the print server
192.168.1.208 HostB
192.168.1.222 HostC          #Uses printer on HostA
192.168.1.101 WCS_Skills
[root@HostA /root]# ping HostC
PING HostC (192.168.1.222) from 192.168.1.206 : 56(84) bytes of data.
Warning: time of day goes back, taking countermeasures.
64 bytes from HostC (192.168.1.222): icmp_seq=0 ttl=255 time=547 usec
64 bytes from HostC (192.168.1.222): icmp_seq=1 ttl=255 time=402 usec

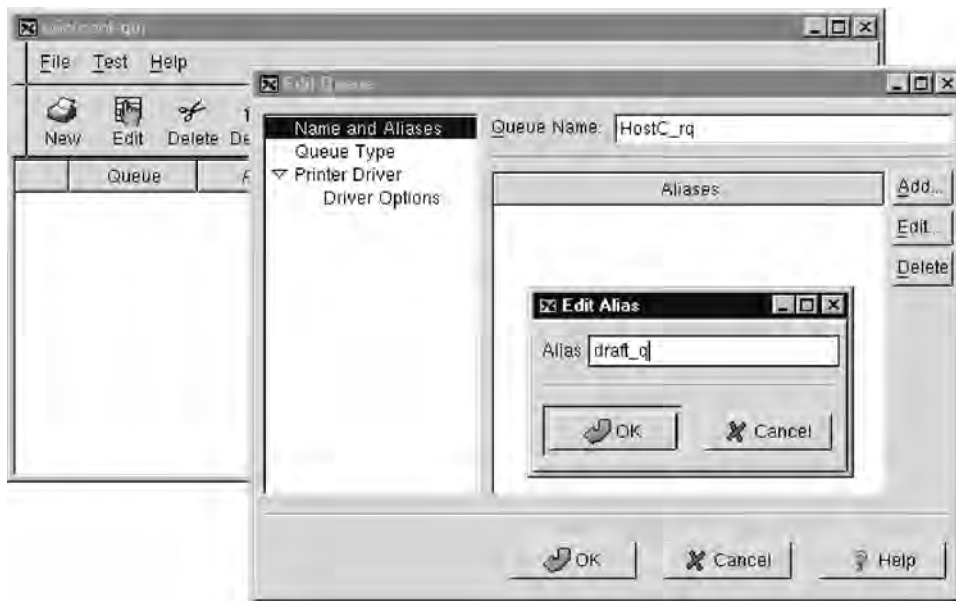
--- HostC ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.402/0.474/0.547/0.075 ms
```

The `/etc/hosts` file on HostC has been updated to include HostA, and a ping test was performed:

```
[root@HostC /root]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
192.168.1.222      HostC #Uses printer on HostA
192.168.1.205      HostB
192.168.1.206      HostA #This system is the print server
[root@HostC /root]# ping HostA
PING HostA (192.168.1.206) from 192.168.1.222 : 56(84) bytes of data.
Warning: time of day goes back, taking countermeasures.
64 bytes from HostA (192.168.1.206): icmp_seq=0 ttl=255 time=1.217 msec
64 bytes from HostA (192.168.1.206): icmp_seq=1 ttl=255 time=437 usec
```

## Step 2. Define the HostC Queue

Log in to the remote system, which in our case is HostC. Make sure that you are the root user, and then execute the `printtool` utility. For clarity, we have defined the Queue Name as `HostC_rq` to remind us that this queue will reside on HostC and will be the remote queue. We have also defined an alias for this queue, as well, to be `draft_rq` (see Figure 6.11).



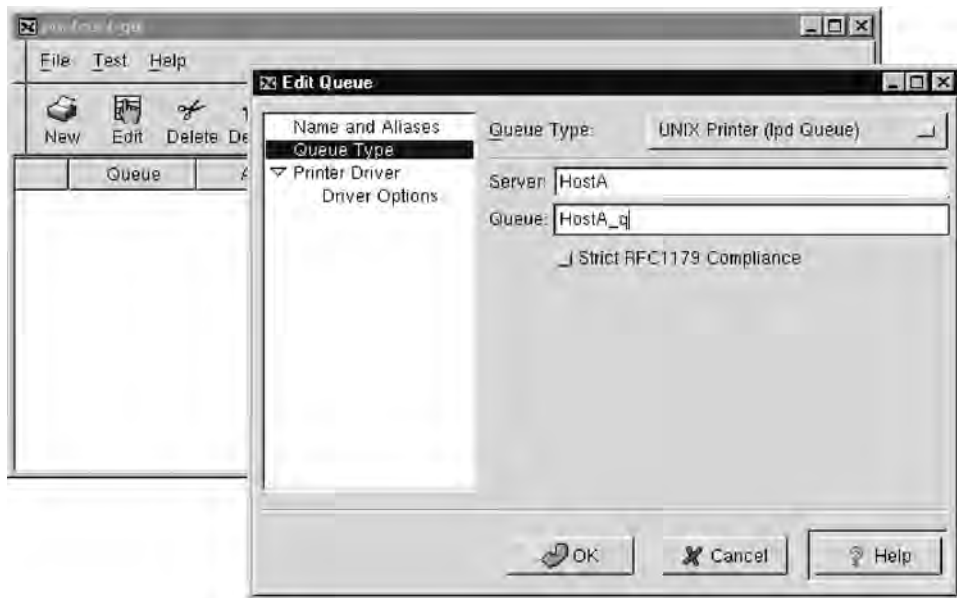
**Figure 6.11** Define the local queue name for the remote printer.

### Step 3. Define the Queue Type as Remote

This time, choose the Remote Unix (lpd) Queue option. Note the reference to lpd in this option. We will be using the lpd daemon to facilitate the printing between the two computer systems. The server that the printer is on is called HostA. For this reason, we had to have our TCP/IP and name resolution working. If the name resolution was not working, then it could not map the server name HostA to an IP address. The queue required here is not the HostC queue but the name of the queue on HostA. The HostA queue was HostA\_q. We could have used the alias here, as well. See Figure 6.12.

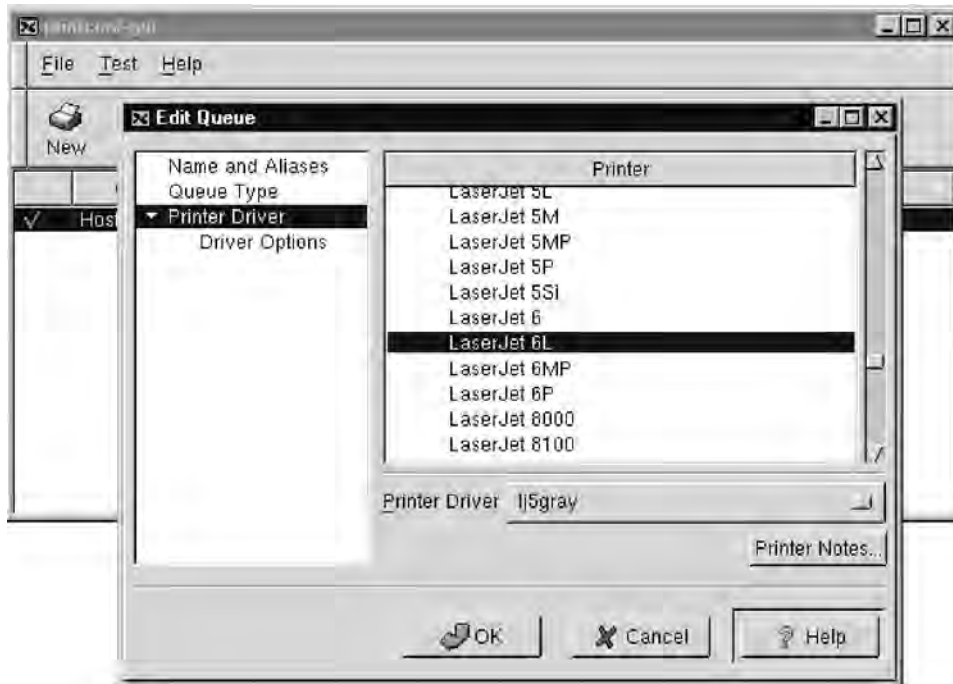
### Step 4. Define the Same Printer Driver

It is important here that you choose the *same* printer driver. Failure to do so might cause some unpredictable results. What you do not want is for the HostC queue to filter the job as a LaserJet 6L with driver lj5gray and the HostA queue to filter the job as a LaserJet 6L using driver lejt4. If you are in a situation where you cannot get the same printer driver enabled at both ends, you can make the receiving queue on HostA a Raw Print Queue device. If you choose to perform this action, then make sure that the users on HostA have a local queue as well to print to that is configured for that



**Figure 6.12** Define the remote server and queue name.





**Figure 6.13** Define a printer driver.

printer. In this configuration, you will have two print queues sending jobs to the printer on HostA. See Figure 6.13.

#### **Step 5. Define the Same Driver Options**

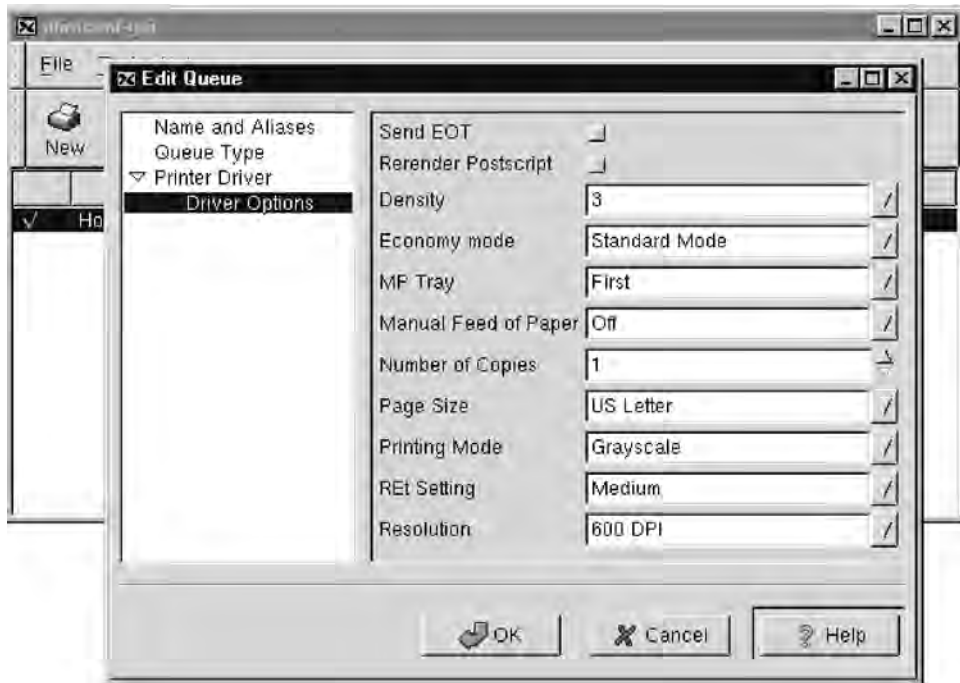
We also recommend that you keep the parameters for the driver options the same, as well (see Figure 6.14).

#### **Step 6. Refresh the lpd Daemon**

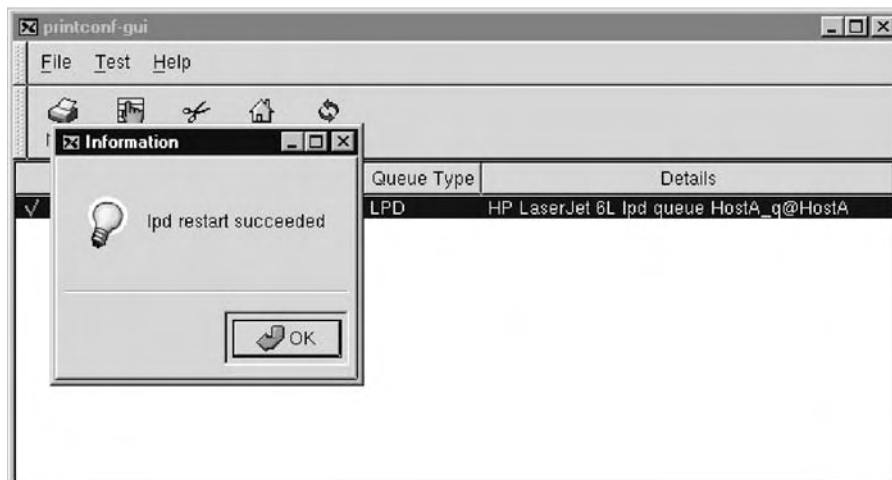
You can use the same method to refresh the lpd daemon here. Nevertheless, before you do that, you should ensure that the `/etc/hosts.lpd` file has an entry for HostA in it so that HostA can access and communicate with this queue over the network. If you have not created the `/etc/hosts.lpd` file yet, then you can wait until after the next step. To refresh the lpd daemon, you can use Apply or use the File drop-down menu Restart lpd. See Figure 6.15.

#### **Step 7. Update the `/etc/hosts.lpd` File**

```
[root@HostC /root]# cat /etc/hosts.lpd
HostA
```



**Figure 6.14** Define additional driver options.



**Figure 6.15** Start or restart the lpd daemon.

You can refresh the `lpd` daemon by using the `printtool`, or you can start and stop the `S60lpd` scripts in the `/etc/rc3.d` directory. There is no difference.

```
[root@HostC /root]# /etc/rc3.d/S60lpd stop
Stopping lpd: [ OK ]
[root@HostC /root]# /etc/rc3.d/S60lpd start
Starting lpd: [ OK ]
```

Note that some system administrators will use the *hosts.equiv* file instead of the *hosts.lpd* file. Only one is necessary. There are two schools of thought on which file to use. The issue is that the *hosts.lpd* file provides specific authorization for the `lpd` daemon only, whereas the *hosts.equiv* file provides general authorization for additional utilities used between systems. Some system administrators do not want to give away more access than they need to just to facilitate a simple printing function. Others say that they require the *hosts.equiv* for a specific general function anyway and do not want an additional file to maintain. Because the *hosts.lpd* file is a configuration file for the `lpd` daemon, its source directory is */etc*, as is the case for all daemon configuration files. Note also that the syntax requirements consist only of the IP address or host name. Remember the following rule for configuration files. You cannot refer to an IP address or host name in a configuration file that is not already resolvable to the network via flat host files, NIS, or DNS.

### **Step 8. Verify Printing**

The following output from the `lpq` command shows that the `HostC_rq` queue has an assigned destination of `HostA_q@HostA`. There are no jobs in the queue just yet, but we can test that next.

```
[root@HostC /root]# lpq
Printer: HostC_rq@HostC 'draft_rq' (dest HostA_q@HostA)
Queue: no printable jobs in queue
Status: job 'cfA489HostC' removed at 19:16:31.031
Printer: HostA_q@HostA 'draft_q'
Queue: no printable jobs in queue
Status: job 'cfA489HostC' removed at 19:16:48.983
```

Users on `HostC` will simply submit their print jobs to the queue called `HostC_rq`. The `lpd` daemon will take care of transporting the job over to the `HostA_q` on `HostA`. Once the job is transferred to `HostA_q`, it will be directed to the local printer in exactly the same way as local jobs on `HostA`. The printing is so fast that we cannot see the jobs in the queue accumulate.

```
[root@HostC /root]# lpr -v grocerylist.txt
[root@HostC /root]# lpq
Printer: HostC_rq@HostC 'draft_rq' (dest HostA_q@HostA)
Queue: no printable jobs in queue
Status: job 'cfA698HostC' removed at 19:58:00.286
Printer: HostA_q@HostA 'draft_q'
Queue: no printable jobs in queue
Status: job 'cfA698HostC' removed at 19:55:26.981
```

Therefore, we are going to put a holdall on the queue called HostA\_q on HostA so that we can confirm that everything is working. Logging into HostA as root, we will execute the following commands:

```
[root@HostA /root]# lpq
Printer: HostA_q@HostA 'draft_q'
Queue: no printable jobs in queue
Status: job 'cfA698HostC' removed at 19:55:26.981
[root@HostA /root]# lpc holdall
Printer: HostA_q@HostA
kill server PID 0 with User defined signal 1
HostA_q@HostA: holdall on
[root@HostA /root]# lpq
Printer: HostA_q@HostA 'draft_q' (holdall)
Queue: no printable jobs in queue
Status: job 'cfA698HostC' removed at 19:55:26.981
```

Logging back into HostC, we will now resubmit print jobs:

```
[vern@HostC vern]$ lpr poetry.txt
[vern@HostC vern]$ lpq
Printer: HostC_rq@HostC 'draft_rq' (dest HostA_q@HostA)
Queue: no printable jobs in queue
Status: job 'cfA757HostC' removed at 20:03:33.839
Printer: HostA_q@HostA 'draft_q' (holdall)
Queue: no printable jobs in queue
Holding: 1 held jobs in queue
Server: no server active
Status: job 'cfA698HostC' removed at 19:55:26.981
Rank  Owner/ID      Class Job Files      Size Time
hold  vern@HostC+757  A    757 poetry.txt    50  20:01:00
```

Now, we can go back to HostA, remove the holdall on the queue, and release the job to print:

```
[root@HostA /root]# lpc noholdall
Printer: HostA_q@HostA
kill server PID 0 with User defined signal 1
```

```
HostA_q@HostA: holdall off
[root@HostA /root]# lpq
Printer: HostA_q@HostA 'draft_q'
Queue: no printable jobs in queue
Holding: 1 held jobs in queue
Server: no server active
Status: job 'cfA698HostC' removed at 19:55:26.981
Rank   Owner/ID          Class Job Files      Size Time
hold   vern@HostC+757     A    757 poetry.txt    50 20:01:00
[root@HostA /root]# lpc release all
Printer: HostA_q@HostA
HostA_q: selected 'vern@HostC+757'
kill server PID 0 with User defined signal 1
HostA_q@HostA: started
```

Now, you should see all the jobs flow through.



# Adding, Replacing, and Upgrading RPM Packages

This chapter is dedicated to RPM packages. RPMs are software packages that can simplify a system professional's work. We begin by defining RPM and RPM packages and showing you how to manage and query them.

We discuss how to determine whether you need to add RPM packages to your system or upgrade the ones already there, and we provide installation and upgrade instructions. We focus on two examples: one where you have just installed a new system and you wish to apply the most recent maintenance to any RPMs that are outdated, and the other uses the XFree86 RPM packages as a prime example (because video is a common issue).

Later in the chapter, we install XF86Setup, a video configuration utility for older RedHat distributions V 6.2 and older, by using RPM packages. XF86Setup is an excellent video configuration utility that has worked well in the past. For some reason, they have discontinued support for XF86Setup and have replaced it with Xconfigurator and xf86cfg. Due to popular request, we have left this section untouched from our previous edition. The reason is because some documentation tells you that you have four alternatives available to configure and set up your video:

- Xconfigurator
- Xf86cfg

- XFree86Setup
- Edit /etc/X11/XF86Config manually

And, at the end of this chapter, we have documented how you add the X windows to a system that has no X on it at all.

## RPM Definition and Package Management

---

RPM is an open packaging system that software vendors, developers, and others use to organize and deliver binary, source, configuration, and other files to users, programmers, and administrators. The mechanism that this system uses to organize and deliver files is the RPM package. An RPM package typically contains binary, executable code and program files. They also have relevant configuration and documentation lines that can be read by using the `rpm` utility. They are an effective way to distribute program updates and upgrades.

The RPM system tracks in the RPM database all packages and associated files installed on a machine with the `rpm` utility. The `rpm` utility is a powerful tool you use to install, query, verify, update, erase, and build software packages that conform to a specific RPM format. For example, you could use the `rpm` utility to query the RPM database to determine whether you have a particular package on your system, whether it is installed, whether it should be upgraded, and so on. The `rpm` utility is executed with the `rpm` command and is available in both GUI and command-line versions.

The RPM database and binary files are built right into a distribution. This feature is relevant because there are dependencies among the packages themselves. These dependencies result in shared code. For example, in the new 2.2.X Linux kernels, the GNU standard C library is referred to specifically as `glibc` or `libc6` code. All executables for this kernel version have been linked against this library. Therefore, if you attempt to install binary files from a different distribution, chances are that they will not work—at least, until you install the `libc5` library for backward compatibility.

Here is an example of the syntax you can use to install an additional XFree86 RPM package onto your system. The command and its options are explained later in the chapter.

```
rpm -ivh XFree86-76dpi-fonts.rpm
```

Prior to upgrading or installing any RPM package, the `rpm man` page should be mandatory reading. In particular, be sure that you understand

how to uninstall and remove packages by using the `rpm` utility, should the RPM package prove not to be what you need.

## RPM Naming Convention

The RPM package-naming convention is the key to package identification. Each package name consists of `Name-version.arch.rpm`, where `Name-version` is the current release of that package and `arch` is the platform designation, such as `Intel`. The standard for Intel is `i386`, although some kernel packages might indicate `i586` or `i686` (and so on).

```
# ls | less
XFree86-100dpi-fonts-3.3.6-20.i386.rpm
XFree86-3.3.6-20.i386.rpm
XFree86-3DLabs-3.3.6-20.i386.rpm
XFree86-75dpi-fonts-3.3.6-20.i386.rpm
XFree86-8514-3.3.6-20.i386.rpm
```

The packages you install must match the versions of the shared libraries available on that machine. If you try to install an RPM package that does not match the shared libraries, you will see an error on the screen. At this point, you could “force” the install. In this case, however, you must be sure that you are not affecting other RPM packages. Of course, it would be proper form to back up a system first before applying this kind of technique.

## SRPM versus RPM

An SRPM file contains the original source used to build the RPM binaries. It is of interest to developers who need to modify and compile RPM packages for their systems. Many programmers will make an SRPM file available to other programmers to facilitate the free software development philosophy within the Linux community. The RPM binaries are the files that system administrators need to apply to the system.

## Determining A Package’s Function and Contents

Linux users are often curious about an RPM package’s function. You can use the `rpm` utility to query an RPM package about its function, as seen in the following example:



```
# rpm -qip kernel-2.2.14-5.0.i586.rpm
Name : kernel Relocations: (not relocateable)
Version : 2.2.14 Vendor: Red Hat Inc.
Release : 5.0 Build Date: Wed Jun 2 30 07:09:08
1999
Install date: (not installed)
Build Host : porky.devel.redhat.com
Group : System Environment/Kernel
Source RPM : kernel-2.2.14-5.0.src.rpm
Size : 12145883
License : GPL
Packager : Red Hat Inc.
<http://bugzilla.redhat.com/bugzilla>
Summary : The Linux kernel (the core of the Linux operating
system).
Description
The kernel package contains the Linux kernel (vmlinuz), the
core of your Red Hat Linux operating system. The kernel
handles the basic functions of the operating system:
memory allocation, process allocation, device input
and output, etc.
```

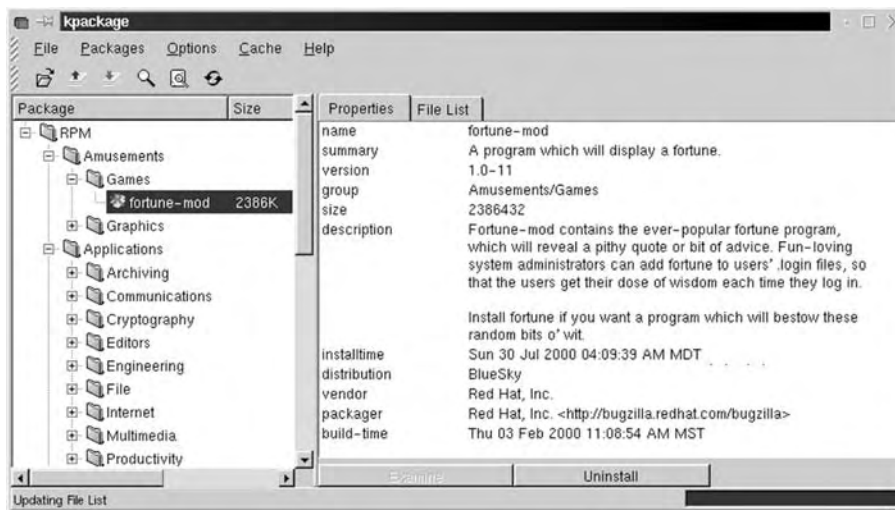
The rpm syntax is as follows:

```
rpm -qip <RPM package-name>
```

The rpm query options used in this section are as follows:

- q    Use the query option specifically to gather package information. Without the -q option, you would get different and unsatisfactory information because the subsequent options would mean entirely different specifications.
- i    When used with the -q option, -i asks rpm to display package information, including the name, version, and package description. Note that if the -q option is not specified along with the -i option, you will get different and unsatisfactory results with the rpm command.
- p    When used with the -q option, the -p option tells rpm that the package is not yet uninstalled.

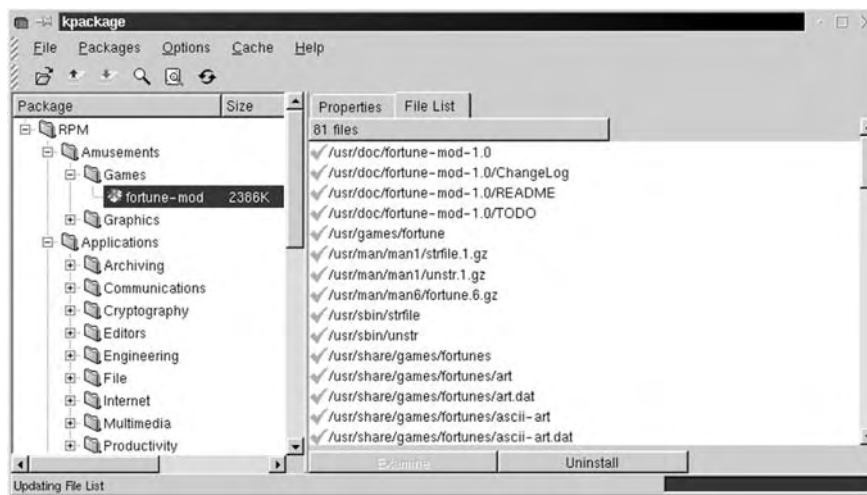
**TIP** Use the **Tab** key for command completion. Type the first part of a command, press **Tab**, and the shell might be able to fill in the rest of the command if you have typed the entire command previously and the shell still remembers it.



**Figure 7.1** Package information displayed from a GUI.

You can examine RPM packages by using a GUI, as shown in Figures 7.1 and 7.2. Here, we examine the function of a games package and the files associated with it.

To practice, query the system for every RPM package that is of interest to you. You will see this utility at work again in Chapter 8, “Upgrading the



**Figure 7.2** File information for an RPM package displayed from a GUI.

Kernel by Using RPM Packages,” when we check and verify the RPM package for a new kernel.

## Applying Maintenance or Updating a Newly Installed System

---

It does not matter where you have acquired the Linux CD-ROM or code. By the time you install it, the code is outdated. We are going to show you how to bring your newly installed system up to the most current level. The following instructions assume that you have just installed a Red Hat system. Our example is V7.1, but these instructions will work for any version. You must also have connectivity to the Internet, because we will use the ncFTP tool to connect to Red Hat’s site to access the updates and download the update files. Then, we are going to examine the update files to see which ones we wish to apply. We do not recommend that you arbitrarily apply everything. This technique could break more than it could fix. Depending on what time distribution you are accessing, the files will differ. For example, a month from now we would expect some additional files and newer files to appear on this site. This factor makes this task a regular job in the life of a system administrator.

We are also recommending that you back up the system at this point. In the event that upgrading these packages causes issues, you can always get back to this point.

### Step 1. Prepare a Directory to Hold the Updates

It is considered best practice to separate your updates from all other files. For example, we have created a separate directory called */LinuxV71Downloads* for these files on a disk where there is adequate disk space available.

```
[root@HostA /]# ls -ld LinuxV71Downloads
drwxr-xr-x 2 root root 4096 Jun 25 16:56 LinuxV71Downloads
```

### Step 2. FTP the Update Files

Using the ncFTP tool, we downloaded the updates directly from RedHat’s site into the *LinuxV71Downloads* directory. Normally, we would not download all the files—but in this case, there were not many of them. Normally, you would only download those that you are interested in updating. First, you do not want to store what you do not need, and second, you can keep this information as a reference for what has already been downloaded.

Some system administrators will download updates into a directory that has the original RPMs in it, but we find it gets too large and it is no longer obvious what is new and what was original. It is also a good idea to write all of this information down in case you have to rebuild this system or create another one just like it.

Notice how all the different distributions have their respective directory. Navigate down to the i386 directory where your files are. Go specifically to the correct directory for your distribution, and list the contents:

```
[root@HostA /]# cd LinuxV71Downloads
[root@HostA /LinuxV71Downloads]# ncftp updates.redhat.com
NcFTP 3.0.2 (October 19, 2000) by Mike Gleason (ncftp@ncftp.com).
Connecting to 63.240.14.69...
"Red Hat FTP server ready. All transfers are logged, please have a nice
day."
ncftp / > ls
1.0/          3.0.3/        5.0/          6.1/          7.1/          lib/
1.1/          4.0/          5.1/          6.2/          bin/          other_prod/
2.0/          4.1/          5.2/          6.2EE/        current@      usr/
2.1/          4.2/          6.0/          7.0/          etc/          var/
ncftp / > cd 7.1/en/os/i386
ncftp / > ls
arts-2.1.2-1.i386.rpm          mgetty-voice-1.1.25-5.i386.rpm
cpp-2.96-85.i386.rpm          minicom-1.83.1-8.i386.rpm
devfsd-2.4.3-12.i386.rpm      mount-2.11b-3.i386.rpm
gcc-2.96-85.i386.rpm          mouseconfig-4.22-1.i386.rpm
gcc-c++-2.96-85.i386.rpm      netscape-common-4.77-1.i386.rpm
gcc-chill-2.96-85.i386.rpm    netscape-communicator-4.77-
                                1.i386.rpm
gcc-g77-2.96-85.i386.rpm      netscape-navigator-4.77-1.i386.rpm
gcc-java-2.96-85.i386.rpm     samba-2.0.8-1.7.1.i386.rpm
gcc-objc-2.96-85.i386.rpm     samba-client-2.0.8-1.7.1.i386.rpm
gftp-2.0.8-1.i386.rpm         samba-common-2.0.8-1.7.1.i386.rpm
gnupg-1.0.6-1.i386.rpm        samba-swat-2.0.8-1.7.1.i386.rpm
kdelibs-2.1.2-1.i386.rpm      SysVinit-2.78-17.i386.rpm
kdelibs-devel-2.1.2-1.i386.rpm up2date-2.5.4-1.i386.rpm
kdelibs-sound-2.1.2-1.i386.rpm up2date-gnome-2.5.4-1.i386.rpm
kdelibs-sound-devel-2.1.2-1.i386.rpm Xconfigurator-4.9.29-1.i386.rpm
kernel-2.4.3-12.i386.rpm      XFree86-3DLabs-3.3.6-38.i386.rpm
kernel-BOOT-2.4.3-12.i386.rpm XFree86-8514-3.3.6-38.i386.rpm
kernel-doc-2.4.3-12.i386.rpm  XFree86-AGX-3.3.6-38.i386.rpm
kernel-headers-2.4.3-12.i386.rpm XFree86-FBDev-3.3.6-38.i386.rpm
kernel-source-2.4.3-12.i386.rpm XFree86-Mach32-3.3.6-38.i386.rpm
krb5-devel-1.2.2-5.i386.rpm   XFree86-Mach64-3.3.6-38.i386.rpm
krb5-libs-1.2.2-5.i386.rpm    XFree86-Mach8-3.3.6-38.i386.rpm
krb5-server-1.2.2-5.i386.rpm  XFree86-Mono-3.3.6-38.i386.rpm
krb5-workstation-1.2.2-5.i386.rpm XFree86-P9000-3.3.6-38.i386.rpm
libstdc++-2.96-85.i386.rpm    XFree86-S3-3.3.6-38.i386.rpm
```

```
libstdc++-devel-2.96-85.i386.rpm      XFree86-S3V-3.3.6-38.i386.rpm
losetup-2.11b-3.i386.rpm              XFree86-SVGA-3.3.6-38.i386.rpm
LPRng-3.7.4-23.i386.rpm               XFree86-VGA16-3.3.6-38.i386.rpm
mgetty-1.1.25-5.i386.rpm              XFree86-W32-3.3.6-38.i386.rpm
mgetty-sendfax-1.1.25-5.i386.rpm      xinetd-2.1.8.9pre15-2.i386.rpm
mgetty-viewfax-1.1.25-5.i386.rpm      ypbind-1.7-8.i386.rpm
ncftp / > binary
ncftp / > mget *
.....
```

### **Step 3. Determine Which Packages to Upgrade**

There is no simple answer to the question, other than research and good utilization of the Linux forums. There are two schools of thought here, and both are good. One is that you do not upgrade anything until there is a problem. The other is to upgrade any file set that you already have on the system to the newer one so that there will be no problems. In either case, you would not add a package just to upgrade it. If you do not have it on your system already, then do not worry about it.

For example, the first package that we see is the `xinetd` package. We know that this package is a major player for all the Internet daemons. We call these *super daemons*. We can check this function by querying the package before we upgrade it:

```
# rpm -qip xinetd-2.1.8.9pre15-2.i386.rpm
Name       : xinetd
Relocations : (not relocateable)
Version    : 2.1.8.9pre15
Vendor     : Red Hat, Inc.
Release    : 2
Build Date : Mon 04 Jun 2001 02:19:48 PM MDT
Install date: (not installed)
Build Host  : porky.devel.redhat.com
Group      : System Environment/Daemons
Source RPM  : xinetd-2.1.8.9pre15-2.src.rpm
Size       : 216452      License: Distributable (BSD-like)
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL         : http://synack.net/xinetd/
Summary     : A secure replacement for inetd.
Description :
```

Xinetd is a secure replacement for `inetd`, the Internet services daemon. Xinetd provides access control for all services based on the address of the remote host and/or on time of access and can prevent denial-of-

access attacks. Xinetd provides extensive logging, has no limit on the number of server arguments, and you can bind specific services to specific IP addresses on your host machine. It also enables information of a service to be in a separate file.

The other daemons that we will look at upgrading on our system are as follows:

```
LPRng-3.7.4-23.i386.rpm
mount-2.11b-3.i386.rpm
mouseconfig-4.22-1.i386.rpm
netscape-common-4.77-1.i386.rpm
netscape-communicator-4.77-1.i386.rpm
netscape-navigator-4.77-1.i386.rpm
SysVinit-2.78-17.i386.rpm
Xconfigurator-4.9.29-1.i386.rpm
xinetd-2.1.8.9pre15-2.i386.rpm
```

#### Step 4. Confirm Current Versions of Packages

Confirm the version of each package on the system before proceeding. You perform this task for each package. This idea is good because the syntax for upgrade will not install in the event that the base package is not there. The only issue that we see is that the netscape-navigator-4.77-1.i386.rpm package is not on the system. We will install the package and not upgrade it accordingly.

```
# rpm -qa | grep LPRng
LPRng-3.7.4-22
# rpm -qa | grep mount
mount-2.10r-5
# rpm -qa | grep mouseconfig
mouseconfig-4.21-1
# rpm -qa | grep netscape
netscape-communicator-4.76-11
netscape-common-4.76-11
# rpm -qa | grep SysVinit
SysVinit-2.78-15
# rpm -qa | grep Xconf
Xconfigurator-4.9.27-1
# rpm -qa | grep xinetd
xinetd-2.1.8.9pre14-6
```

**Step 5. Proceed to Upgrade the Packages**

Repeat this step for every package that you are going to upgrade:

```
# rpm -Uvh LPRng-3.7.4-23.i386.rpm
Preparing...      ##### [100%]
 1:LPRng          ##### [100%]
```

To install a new (that is, new to our system) RPM package, use the following:

```
# rpm -ivh netscape-navigator-4.77-1.i386.rpm
```

Command-line options are listed as follows:

- U      Upgrade the RPM package.
- i      Install the RPM package.
- v      Verify, meaning compare information about the installed files with information about the files found in the original package stored in the RPM database, and display any discrepancies.
- h      Hash, meaning print 50 hash marks as the package archive is unpacked, to indicate its progress.

Note that the *i* option here means to install the RPM package. Previously, when *rpm* was used to query package contents in the section called *Determining A Package's Function and Contents*, we saw how the *i* option meant something completely different when it was used in conjunction with the *q* option. Then, it was only used to find information about a specified RPM package.

Upgrading and installing, in that order, should prevent any potential prerequisite problems among the RPM packages.

On rare occasions, you might experience conflicts between packages; the shell might return an error message that tells you there is a conflict. In this case, you might have to force the installation of a new package. To do so, use the following syntax:

```
rpm --force <RPM package-name>
```

You should force an installation only if you are sure that the installation will not adversely affect a package dependency. That is, if one package depends on a particular version of another package or shares its code base, you should not upgrade the second package by forcing its installation.

If you have many RPM packages to install, consider using a wildcard. The `rpm` command takes wildcard arguments for filenames. An example is as follows:

```
rpm -ivh *rpm
```

### **Step 6. Query the RPM Database to Confirm the Upgrades**

Use the `rpm` command to confirm the upgraded packages. The ultimate test, however, will be conducted over the next week as the system performs. If problems occur, you can try to back out the package and replace it with the old one. There is no process for downgrading. Remove it and install the other one. Of course, this procedure creates a nightmare of dependencies and requisites. So, do not be surprised if you have to resort to restoring the system from your backup as mentioned in Step 1. Be careful and thoughtful about this process.

## **Replacing and Upgrading the XFree86 Package**

---

Among the most common periodic procedures, a system is the replacement and upgrading of RPM packages for video support. To demonstrate, we will replace the XFree86 package. We chose this action for two reasons. First, the XFree86 package that comes with the distribution has only the most basic drivers, rather than all those you might find necessary. Second, the newer XFree86 package has additional support that can help to fine-tune the video system. This situation is particularly true of the older distributions prior to (and not including) RedHat V7.0.

We recommend that you become familiar with a trustworthy Web site and consistently use it as a source for your RPM packages. The major reason for doing so is to make it easier to track new releases by becoming familiar with the layout of the site. We recommend that you bookmark the following two Web sites: [www.rpm.org](http://www.rpm.org) and <ftp://ftp.muug.mb.ca/mirror/redhat/redhat/updates>.

The next four steps show you how to use RPM packages to add video support to a system that currently does not have the necessary RPM packages in place. For example, the standard VGA, SVGA, S3, S3V, and Mach drivers ship with the Linux distribution. You might need the AGX or I128 server for your system instead, however, in which case you might be



expected to locate the necessary RPM packages, download them, and add them to the system.

What you have to do is first decide what level of support you need, which is version specific. For example, after having installed RedHat V7.1, our XFree86 RPMs were at level 4.0.3-5. We tried to back a couple of our systems off this level so that we could use the 3.3.8 with the XF86Setup utility, but we found it to be almost impossible. We do not recommend that you try it. We ended up restoring the systems from backup. There appears to be no graceful way to revert to an earlier version. So, say goodbye to that most useful tool. The advice that we give is to always visit either RedHat or one of the mirror sites to see what the current updates are. For example, we are currently running RedHat V7.1. Therefore, we used RedHat's ncFTP tool to connect to updates.redhat.com and download all of the packages.

### **Step 1. Identify RPM Packages to Replace**

Sometimes more than one RPM package is involved when installing or upgrading a utility or feature. To add a new feature, you often have to upgrade an existing RPM package, as well. Therefore, one of the first things that you have to do is meticulously record the versions of the RPM packages already installed on the system by using the rpm utility to query the RPM database. Then, compare those RPMs and their respective versions to the ones that you have found on the Internet. The versions and other requirements are usually buried in the *README* file.

Two examples appear next. Note how we use rpm to query the RPM database for the XFree86 and XFree86-75dpi-fonts RPM packages:

```
# rpm -q XFree86
XFree86-3.3.6-20
# rpm -q XFree86-75dpi-fonts
XFree86-75dpi-fonts-3.3.6-20
```

You have probably noticed that querying a specific package requires that you know the name of the package. If you are uncertain of a package's name, you find it by formulating a command by using rpm to query all packages and grep to display only the packages containing, for example, the text string free. Thus, your alternative command would be as follows:

```
# rpm -qa | grep -i free
XFree86-3.3.6-20
XFree86-libs-3.3.6-20
XFree86-xfs-3.3.6-20
```

```
XFree86-75-dpi-fonts-3.3.6-20
XFree86-Mach64-3.3.6-20
XFree86-devel-3.3.6-20
Freetype-1.3.1-5
XFree86-SF86Setup-3.3.6-20
```

In this command line, `q` stands for “query” and `a` stands for “all installed packages on the system.” As for the `grep` command, the `i` option means “ignore case of text string.”

If your text string is found among the package names, command results will list one or more package names found on the system. If your text string query is not even close, then the command will return nothing and you will have to try other ideas.

In the current example, as you can see, the command tips you off to the existence of several RPM packages whose names include “free,” such as the XFree86 packages sought in the previous two example queries.

## Step 2. Obtain Necessary Files and Place on the System

Organizing your work is recommended. We suggest that you create a `/usr/sysadmin/XFree86` directory in which to place the new RPM packages. This practice will help you maintain an audit trail of what you have added to the system. RPM packages do not have to be placed in any specific directory. They will be added to the system by using the `rpm` command. Thus, they can be deleted from this directory after they are installed.

We assume that you have already acquired the following files via `ftp` or copied the RPM packages from their source. Our source for these RPM packages was `www.rpm.org` and `ftp://updates.redhat.com`.

```
# ls /usr/sysadmin/XFree86
XFree86-100dpi-fonts-3.3.6-20.i386.rpm
XFree86-3.3.6-20.i386.rpm
XFree86-3DLabs-3.3.6-20.i386.rpm
XFree86-75dpi-fonts-3.3.6-20.i386.rpm
XFree86-8514-3.3.6-20.i386.rpm
XFree86-AGX-3.3.6-20.i386.rpm
XFree86-FBDev-3.3.6-20.i386.rpm
XFree86-I128-3.3.6-20.i386.rpm
XFree86-ISO8859-2
XFree86-ISO8859-2-100dpi-fonts
XFree86-ISO8859-2-75dpi-fonts
XFree86-ISO8859-2-Type1-fonts
XFree86-ISO8859-7
```

.  
.  
.

### **Step 3. Make Notes on Packages to Be Upgraded versus Packages to Be Installed**

You perform upgrades first and then add new RPM packages. If a package that you are upgrading or adding requires another package, then it will advise you. In this case, you will have to locate that package and deal with it first. The following examples show how to upgrade and install an RPM package.

To upgrade an existing RPM package,

```
# rpm -Uvh XFree86-3.3.5-1.6.0.i386.rpm  
# rpm -Uvh XFree86-75dpi-fonts-3.3.5-1.6.0.i386.rpm
```

To install a new (that is, new to our system) RPM package,

```
# rpm -ivh XFree86-XF86Setup-3.3.5-1.6.0.i386.rpm
```

Command-line options are listed as follows:

- U      Upgrade the RPM package.
- i      Install the RPM package.
- v      Verify, meaning compare information about the installed files with information about the files found in the original package stored in the RPM database, and display any discrepancies.
- h      Hash, meaning print 50 hash marks as the package archive is unpacked, to indicate its progress.

Note that the *i* option, here, means to install the RPM package. Previously, we saw how the *i* option meant something completely different when it was used in conjunction with the *q* option. Then, it was only used to find information about a specified RPM.

Package Upgrading and installing, using the specified sequence, should prevent any potential prerequisite problems among the RPM packages.

On rare occasions, you might experience conflicts between packages; the shell might return an error message that tells you there is a conflict. In this case, you might have to force the installation of a new package. To do so, use the following syntax:

```
rpm --force <RPM package-name>
```

You should force an installation only if you are sure that the installation will not adversely affect a package dependency. That is, if one package depends on a particular version of another package or shares its code base, you should not upgrade the second package by forcing its installation.

If you have many RPM packages to install, consider using a wildcard. The `rpm` command takes wildcard arguments for filenames. An example is as follows:

```
rpm -ivh *rpm
```

#### **Step 4. Query the RPM Database to Confirm the Upgrade**

Use the `rpm` command to confirm the installation and upgrade. The ultimate test, however, is to confirm that the application is running correctly without problems. An example command is shown next. You can see that the command is identical to the `rpm` query used in Step 1.

```
# rpm -qa | grep -i free
freetype-1.2-6
XFree86-75dpi-fonts-3.3.5-1.6.0
XFree86-libs-3.3.5-1.6.0
XFree86-xfs-3.3.5-1.6.0
XFree86-SVGA-3.3.5-1.6.0
XFree86-VGA16-3.3.5-1.6.0
XFree86-3.3.5-1.6.0
XFree86-XF86Setup-3.3.5-1.6.0
XFree86-100dpi-fonts-3.3.5-1.6.0
XFree86-S3-3.3.5-1.6.0
XFree86-S3V-3.3.5-1.6.0
```

## **Using Packages to Install XF86Setup**

---

In Chapter 5, “Configuring the Video,” we configured the video system with the `Xconfigurator` utility program. This time, you repeat the same exercise using an alternative GUI-like tool called `XF86Setup`, which we mentioned in Chapter 5 but did not discuss in detail. `XF86Setup` has to be installed first, however, and we can do so by using a package. Figure 7.3 shows the `XF86Setup` tool introductory dialog.

The `XF86Setup` tool essentially does the same things as `Xconfigurator`. Both tools create a functional `/etc/XF86Config` file. `XF86Setup`, however, enables you to experiment more easily with alternate video servers. For example, instead of using the standard default SVGA server, you can



**Figure 7.3** XF86Setup tool introductory dialog box.

quickly choose, say, the SV3 server if it is more appropriate to your hardware. Changing from one server to a more appropriate server can make the difference between a mediocre system and a better-functioning, more responsive system.

The next exercise will familiarize you with the installation of this new application that uses RPM packages. It will also provide you with experience in updating and adding new video support to your system by upgrading and installing the XF86Setup software.

## Tutorial

Replacing and upgrading XFree86 packages are probably two of the more common system administration procedures. They are the first items that will require attention on most systems. This exercise is straightforward. These instructions can be used on all versions of Red Hat Linux as long as you ensure that the package versions are consistent with the existing version of both the distribution and the kernel levels. We

demonstrate how to achieve that in the exercise. This particular example is taken from Red Hat Linux Versions 5.0, 5.2, and 6.0 because these particular versions did not have the most current XFree86 RPM packages available (which made it awkward to work with out of the box). By contrast, the Red Hat Linux Version 6.2 came with current XFree86 RPM packages.

The following exercise is by example in the sense that the actual revision numbers on the RPM packages might vary from system level to system level. We encourage you to take good notes as you go along to ensure a successful upgrade or installation.

1. Determine which RPM packages need replacing and their revision levels by querying the RPM database.

```
# rpm -qa | grep -i xfree86
XFree86-3.3.3.1-49
XFree86-75dpi-fonts-3.3.3.1-49
XFree86-libs-3.3.3.1-49
XFree86-xfs-3.3.3.1-49
XFree86-SVGA-3.3.3.1-49
```

**NOTE** The output statements listed in these exercises occurred as a result of the authors' use of the commands with an older version of Linux. Your output will likely be different depending on which version of Linux you have installed and whether it has already been upgraded.

2. Check with a reputable Web site to research and obtain the correct packages. Compare their version numbers and dates to the packages on your CD media. Use whichever ones are newer.

If it turns out that the packages on the Web site are newer than the packages on your CD media, you will have to download them. (In preparing this exercise, we required a copy of the S3 or S3V video server to support the S3 video chipset. So we downloaded several additional packages [seen as follows] from the updates.redhat.com ftp site, and we put them in a directory called `/usr/sysadmin/XFree` on the Linux server that we wished to update.)

3. Proceed to create the directory in which to put the new RPM packages.

```
# mkdir -p /usr/sysadmin/XFree
```

4. Using ftp, copy all the files from `/usr/sysadmin/XFree` onto your server.

```
# ls
XFree86-100dpi-fonts-3.3.5-1.6.0.i386.rpm
XFree86-3.3.5-1.6.0.i386.rpm
XFree86-75dpi-fonts-3.3.5-1.6.0.i386.rpm
XFree86-S3-3.3.5-1.6.0.i386.rpm
XFree86-S3V-3.3.5-1.6.0.i386.rpm
XFree86-SVGA-3.3.5-1.6.0.i386.rpm
XFree86-VGA16-3.3.5-1.6.0.i386.rpm
XFree86-XF86Setup-3.3.5-1.6.0.i386.rpm
XFree86-libs-3.3.5-1.6.0.i386.rpm
XFree86-xfs-3.3.5-1.6.0.i386.rpm
```

5. Make notes on the packages that require updating versus those to be installed. Then, proceed to install and/or upgrade the new packages onto the system by using the appropriate rpm command.

```
# cd /usr/sysadmin/XFree
rpm -Uvh XFree86-3.3.5-1.6.0.i386.rpm
rpm -Uvh XFree86-75dpi-fonts-3.3.5-1.6.0.i386.rpm
rpm -Uvh XFree86-libs-3.3.5-1.6.0.i386.rpm
rpm -Uvh XFree86-xfs-3.3.5-1.6.0.i386.rpm
rpm -Uvh XFree86-SVGA-3.3.5-1.0.i386.rpm
rpm -ivh XFree96-VGA16-3.3.5-1.6.0.i386.rpm
rpm -ivh XFree86-XF86Setup-3.3.5-1.6.0.i386.rpm
rpm -ivh XFree86-100*
rpm -ivh XFree86-S3*
```

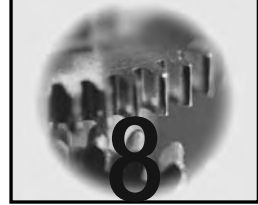
6. Query the packages to ensure that they are placed in the system correctly.

```
# rpm -qa | grep -i xfree
XFree86-75dpi-fonts-3.3.5-1.6.0
XFree86-libs-3.3.5-1.6.0
XFree86-xfs-3.3.5-1.6.0
XFree86-SVGA-3.3.5-1.6.0
XFree86-VGA16-3.3.5-1.6.0
XFree86-3.3.5-1.6.0
XFree86-XF86Setup-3.3.5-1.6.0
XFree86-100dpi-fonts-3.3.5-1.6.0
XFree86-S3-3.3.5-1.6.0
XFree86-S3V-3.3.5-1.6.0
```

7. Execute the new XF86Setup utility, and reconfigure your system video parameters.

```
# XF86Setup
```

For technical details on solving configuration issues, refer to Chapter 5 for how to configure the video.



# Upgrading the Kernel by Using RPM Packages

In Chapter 7, we covered the basics of RPM packages. In this chapter, we upgrade a Linux operating system kernel from 2.4.2-2 to 2.4.3-12 by using RPM packages. We also provide systematic suggestions for handling the upgrade. Finally, we explain what RAM disk support is and how to modify LILO to boot from the new kernel.

You might decide to modify these suggestions to suit your needs. You can upgrade any kernel by using these basic instructions, however. Only those steps related to the basic preparation and administration of the system must be planned, and they can change from system to system.

Generally, the advantages of using RPM packages are that no compiling is necessary and that tedious little details are automated. We eventually illustrate the operational advantages of performing a kernel upgrade. The upgrade uses RPM packages throughout the chapter. A true comparison, however, will come after you perform the “scratch” kernel upgrade in the exercise in Chapter 10, “Installing a New Kernel by Building a New Source Tree.” As a system administrator, you might have cause to perform one upgrade or another (or perhaps both). After performing the exercises in both this chapter and Chapter 10, we think that you will agree with us that



using packages is much safer and more accurate. There are fewer opportunities for errors. Some of the material in this chapter is a restatement of Chapter 7. Because we focus on the operating system kernel, however, the steps we outline will be more detailed with respect to operating practices, the directory structure, and so on.

## **Why Upgrade or Patch the Kernel?**

---

The three main reasons for upgrading or patching the operating system kernel are as follows:

- To obtain support for new hardware
- To obtain updated drivers
- To respond to or remedy security concerns

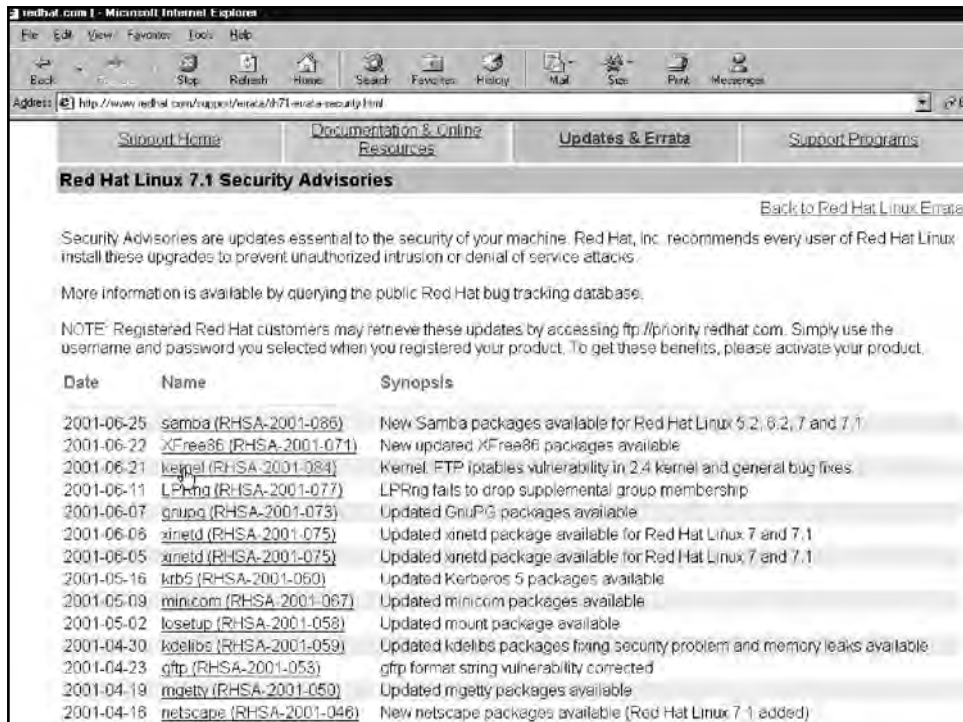
In reality, these are the same reasons for upgrading or patching any application. We hope to convince you in this chapter that the use of RPM packages is the best way to perform upgrades. If you are not convinced by the end of this chapter, we think you will be totally convinced by the end of Chapter 10, where we show you how to upgrade a kernel from scratch with no RPMs.

## **A Typical Security Advisory**

Examples of the first two reasons for upgrading or patching an application (obtaining new hardware support and updating drivers) appeared in Chapter 7. You might ask, “What about the third reason—dealing with security?”

Let’s assume that you are running Red Hat 6.2, which shipped with the Linux kernel version of 2.2.14-5. At one time, a document appeared at [www.redhat.com/support/errata/rh71-errata-security.html](http://www.redhat.com/support/errata/rh71-errata-security.html) that described a serious security issue. The document indicated that the security issue would be fixed if users upgraded to a kernel level of 2.4.3-12.

Fortunately, the document also stated that new packages to correct the vulnerability were available from <ftp://updates.redhat.com>. In this situation, you would obtain the new kernel RPM packages from the aforementioned FTP site and apply them to the system. See Figure 8.1 for an example of a security advisory.



**Figure 8.1** An example of a security advisory.

It is the responsibility of the system administrator to visit this site on a regular basis. It is best practice to check for any security advisories that pertain to the operating system. Each security advisory is different. This figure presents the explanation at the beginning of the security advisory and provides all the links and instructions for applying the necessary RPM packages.

## Suggested Upgrade Procedure for Red Hat V7.X

There are many ways to upgrade your system by using RPM packages. In the following section, we describe a 10-step procedure. Because there are other approaches, we suggest that you use our 10 steps only as a guide. Depending on how your system is configured and how it acts and interacts, you might have to modify the steps a bit.

**WARNING** Regardless of how you execute an upgrade, we definitely recommend that you back up the entire system before proceeding (particularly a production system that has data on it).

### Step 1. Check Disk Space

The kernel will be built inside the `/usr/src/linux/arch/ix86/boot` directory. If you have a separate file system partition for `/usr`, make sure that there is at least an extra 75MB of additional space in it. This amount of space should also be enough for the new RPM packages. The kernel file itself will eventually have to be copied to the `/boot` directory, and the modules will be built in the `/lib` directory. These two items will use approximately 10MB to 15MB inside the root (`/`) file system.

The `df -h` command displays disk usage according to each file system partition as follows. The `h` flag is for human readable form.

```
[root@HostA /root]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda7        4.2G  1.1G  3.0G  26% /
/dev/hda1         53M   3.5M   46M   7% /boot
/dev/hda5       988M   36M  902M   4% /usr/local
```

Refer to the man pages for additional information about the `df` command.

### Step 2. Identify Packages Already Installed

Next, determine which packages are already installed on your system. Make a note of them, and then compare them to the packages required to upgrade the kernel. Appearing as follows is an example query to identify installed packages. You will also want to run the `uname -a` command to see which kernel you are currently using. Notice in the output where it states that this kernel is the i686 kernel. Be careful, because you could downgrade your kernel here. Our goal here is to replace the old i686 kernel with a newer i686 kernel and not a newer i386 kernel. The i386 kernel would work, but it would not perform nearly as well. We address this issue in Chapter 10.

```
[root@HostA /V71]# rpm -qa | grep kernel
kernel-headers-2.4.2-2
kernel-2.4.2-2
kernel-source-2.4.2-2
[root@HostA /V71]# uname -a
Linux HostA 2.4.2-2 #1 Sun Apr 8 20:41:30 EDT 2001 i686 unknown
```

The exact information regarding the RPM packages that are required to upgrade the kernel was not readily available. There was information on HOWTO, but it was not decisive. We visited the `ftp://updates.redhat.com` site to download the most current stable version of the Linux kernel from the `i686` directory. This directory is the same location where we downloaded the new packages in Chapter 7 but over one directory. The FTP dialog is as follows:

```
[root@HostB /root]# mkdir /usr/local/V71
[root@HostB /root]# ln -s /usr/local/V71 /V71
[root@HostA /V71]# ncftp updates.redhat.com
NcFTP 3.0.2 (October 19, 2000) by Mike Gleason (ncftp@ncftp.com).
Connecting to 63.240.14.72...
"Red Hat FTP server ready. All transfers are logged, please have a nice
day."
Logging in...
Guest login ok, access restrictions apply.
Logged in to updates.redhat.com.
ncftp / > ls
1.0/      3.0.3/    5.0/      6.1/      7.1/      lib/
1.1/      4.0/      5.1/      6.2/      bin/      other_prod/
2.0/      4.1/      5.2/      6.2EE/    current@  usr/
2.1/      4.2/      6.0/      7.0/      etc/      var/
ncftp / > cd 7.1/en/os/i686
ncftp /7.1/en/os/i686 > ls
kernel-2.4.3-12.i686.rpm          kernel-smp-2.4.3-12.i686.rpm
kernel-enterprise-2.4.3-12.i686.rpm
ncftp /7.1/en/os/i686 > binary
ncftp /7.1/en/os/i686 > mget kernel-2.4.3-12.i686.rpm
kernel-2.4.3-12.i686.rpm:          9.26 MB   12.47 kB/s
```

You can investigate existing packages by using the `rpm` command. This command can be used to query the packages already installed on the system. Make sure that you note their version levels, as well. Naturally, you will also wish to make notes regarding the RPM packages required that are not on your system.

In Chapter 7, we discussed how the `rpm` utility could help verify what an RPM package can do for you. Here, it is important that we know that our kernel upgrade package is the right one. As illustrated in Chapter 7, the `rpm` syntax is as follows:

```
rpm -qip RPM <package-name>
```

The `rpm` query options are reiterated next.

- q Use the query option specifically to gather package information. Without the -q option, you would get different and unsatisfactory information because the subsequent options would mean entirely different specifications.
- i When used with the -q option, -i asks rpm to display package information, including the name, version, and package description. Note that if the -q option is not specified along with the -i option, you will get different and unsatisfactory results from the rpm command.
- p When used with the -q option, -p tells rpm that the package is not yet installed.

We will use rpm with these options to verify that we have the right package.

```
[root@HostA /V71]# rpm -qip kernel-2.4.3-12.i686.rpm
Name       : kernel           Relocations: (not relocateable)
Version    : 2.4.3           Vendor: Red Hat, Inc.
Release    : 12
Build Date: Fri 08 Jun 2001 01:34:00 PM MDT
Install date: (not installed)
Build Host: porky.devel.redhat.com
Group      : System Environment/Kernel
Source RPM: kernel-2.4.3-12.src.rpm
Size       : 23823653          License: GPL
Packager   : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Summary    : The Linux kernel (the core of the Linux operating system)
Description:
The kernel package contains the Linux kernel (vmlinuz), the core of your
Red Hat Linux operating system. The kernel handles the basic functions
of the operating system: memory allocation, process allocation, device
input and output, etc.
```

Thus far, it appears that we have found the right package.

### **Step 3. Organize Packages and Perform an Integrity Check**

Checking the integrity of the files before they are applied to the system is recommended. Again, you can accomplish this task easily by using the rpm command. The command shown in the following example will check the internal consistency of the files to ensure that there was no damage done during the download process, as seen in the following example:

```
[root@HostA /V71]# rpm -K --nopgp kernel*
error: kernel-2.4.3-12.i386.rpm: rpmReadSignature failed
kernel-2.4.3-12.i686.rpm: md5 GPG NOT OK
kernel-headers-2.4.3-12.i386.rpm: md5 (GPG) OK (MISSING KEYS:
GPG#DB42A60E)
kernel-source-2.4.3-12.i386.rpm: md5 (GPG) OK (MISSING KEYS:
GPG#DB42A60E)
```

Note that the system returns a “failed” message for one of the packages, indicating that it does not match the source. In this case, you should download the package again and rerun the integrity check. In this example, we will download the kernel-2.4.3-12.i386.rpm package again.

#### Step 4. Compare Existing Packages to Downloaded Packages

This step indicates which RPM packages are to be upgraded or installed on the system. Mark all of them off before you begin. In our example, the following will be replaced by the 2.4.3 version of the kernel packages:

```
[root@HostA /V71]# rpm -qa | grep kernel
kernel-headers-2.4.2-2
kernel-2.4.2-2
kernel-source-2.4.2-2
```

Comparing the list of the new RPM packages and the list of the already installed RPM packages, you have to make notes. Specifically, note what RPM packages are to be upgraded and which are to be added so that you can use the proper commands.

#### Step 5. Record the Current Version

The purpose of this step is to check the current version of the kernel. You can check it by looking at the login herald, such as the one shown in the following example:

```
Red Hat Linux release 7.1 (Seawolf)
Kernel 2.4.2-2 on an i686
login:
```

Another way of checking the current version is to use the `uname` command with the `-r` option.

```
[root@HostA /]# uname -r
2.4.2-2
```

A third way to check is to use the `uname` command with the `-a` option.

```
[root@HostA /]# uname -a
Linux HostA 2.4.2-2 #1 Sun Apr 8 20:41:30 EDT 2001 i686 unknown
```

If you are already logged in, use the `uname -a` command to make note of both the version and the processor type that is currently on the machine. In the previous example, the processor is an i686 (that is, some type of Pentium equivalent). If you did not know the processor type, you could simply use the i386, but why downgrade that feature when it is not necessary?

### **Step 6. Create an Emergency Boot Diskette**

The purpose of this step is to provide you with useful information in the event that something goes wrong. If you did not make an emergency boot diskette during installation, this method is the easiest way to create one now. Examine the `/etc/lilo.conf` file and note the kernel image name and version currently being used. In the following example, the file contains `image=/boot/vmlinuz-2.4.2-2`:

```
[root@HostA /V71]# cat /etc/lilo.conf
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
default=linux

image=/boot/vmlinuz-2.4.2-2
    label=linux
    read-only
    root=/dev/hda7
```

Therefore, we can construct the syntax for the `mkbootdisk` command in the following way:

```
# mkbootdisk --device /dev/fd0 2.4.2-2
Insert a disk in /dev/fd0. Any information on the disk will be lost.
Press <Enter> to continue or ^C to abort:
```

Note that there are two dashes before the device option but no space between them. Refer to the man pages for additional information on the `mkbootdisk` command.

## Step 7. Proceed to Install and Upgrade Packages

If you get errors about dependencies, you will have to take care of these first and then return to the upgrade or install procedure. If the two kernel versions are too far apart, you might find that there are conflicts between packages. The shell will return an error message because of using the `rpm` command. You might have to force the install. These concepts, the commands, and their options are discussed in Chapter 7.

If non-kernel packages are involved and are required, it is important to do that first. Then, move on to the kernel-related packages. Although this step is not required for our example upgrade, an example follows.

```
# rpm -Uvh kernel-pcmcia-cs*
```

Install the new kernel source. We recommend that you install and not upgrade the `kernel-source` package. We have experienced some unpredictable results when upgrading as opposed to installing the new kernel source packages. We recommend that you install the new source and delete the old source at a later time.

```
[root@HostA /]# rpm -ivh kernel-source-2.4.3-12.i386.rpm
Preparing... ##### [100%]
 1:kernel-source ##### [100%]
```

Next, upgrade the `kernel-headers` and `kernel` packages.

```
[root@HostA /V71]# rpm -Uvh kernel-headers-2.4.3-12.i386.rpm
Preparing... ##### [100%]
 1:kernel-headers ##### [100%]
[root@HostA /V71]# rpm -Uvh kernel-2.4.3-12.i686.rpm
Preparing... ##### [100%]
 1:kernel ##### [100%]
```

Now, check to see whether the new kernel packages are in place:

```
[root@HostA /root]# rpm -qa | grep kernel
kernel-2.4.3-12
kernel-source-2.4.2-2
kernel-source-2.4.3-12
kernel-headers-2.4.3-12
[root@HostA /root]# ls -l /boot > flist
[root@HostA /root]# awk ' { print $1 " " $2 " " $3 " " $6 " " $7 " " $8 "
" $9 }' flist
total 3416
-rw-r--r-- 1 root Jun 26 15:38 boot.0300
```



```
-rw-r--r-- 1 root Feb 23 12:33 boot.b
-rw-r--r-- 1 root Feb 23 12:33 chain.b
lrwxrwxrwx 1 root Jun 27 09:51 kernel.h
-rw-r--r-- 1 root Jun 27 09:54 kernel.h-2.4.3
drwxr-xr-x 2 root Jun 26 15:16 lost+found
-rw----- 1 root Jun 27 09:58 map
-rw-r--r-- 1 root Feb 23 12:33 message
lrwxrwxrwx 1 root Jun 27 09:54 module-info
-rw-r--r-- 1 root Jun 8 13:32 module-info-2.4.3-12
-rw-r--r-- 1 root Feb 23 12:33 os2_d.b
lrwxrwxrwx 1 root Jun 27 10:00 System.map
-rw-r--r-- 1 root Jun 8 13:32 System.map-2.4.3-12
-rwxr-xr-x 1 root Jun 8 13:32 vmlinuz-2.4.3-12
lrwxrwxrwx 1 root Jun 27 09:54 vmlinuz
-rw-r--r-- 1 root Jun 8 13:32 vmlinuz-2.4.3-12
```

In any event, do not proceed to the next step until this step goes well. In addition, keep in mind that you can use the `rpm` command to either remove or uninstall a bad package. As always, refer to the man pages for additional information on the `rpm` command.

### Step 8. Optional: Create a RAM Disk Support File

The only criterion to meet here is, “Is this a SCSI or laptop computer that requires PCMCIA support?” If so, then you need RAM disk support. The easiest way to determine whether you need RAM disk support is to examine the `lilo.conf` file. There is an additional statement in the `/etc/lilo.conf` file that refers to an initial RAM disk file. That reference will resemble `initrd=/boot/initrd-2.4.3-12.img`. Upon listing the files in the `/boot` directory, you will see the `initrd-2.4.3-12` file. The purpose of this initial RAM disk is to allow a modular kernel, such as Linux 2.2.X, to have access to modules required for the boot process as the system boots up. Thus, it resolves the old catch-22 (or, if you prefer, the chicken-and-egg scenario) where you need a driver to talk to the hardware (such as a SCSI or PCMCIA driver) that is not available (yet) until the disk is up. Essentially, by creating a RAM disk file, you solve this problem by creating an additional pointer (via LILO), which in turn writes to the *master boot record* (MBR). In turn, the MBR will now know to use this file to boot. The necessary modules are linked within this RAM disk file. You will have to create a new `initrd` file for the system to boot from by using the `mkinitrd` command. We do not need this file, but we will go through the procedure to complete the example.

```
[root@HostA /root]# mkinitrd /boot/initrd-2.4.3-12 2.4.3-12
[root@HostA /root]# ls -l /boot | grep initrd
-rw-r--r-- 1 root root 251444 Jun 27 11:06 initrd-2.4.3-12
```

### Step 9. Setting up LILO to Boot from a New Kernel

In this step, you edit *lilo.conf* to point to the new kernel. Remember that if you want to boot from this new kernel all the time, place it as the first stanza in the file. That will make it the default boot item. In addition, remember to add the *initrd* file if you choose to use one. An example of a *lilo.conf* file is as follows:

```
[root@HostA /root]# cat /etc/lilo.conf
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
default=linux

image=/boot/vmlinuz-2.4.3-12
    label=linux
    initrd=/boot/initrd-2.4.3-12
    read-only
    root=/dev/hda7
```

Then, update the MBR by running the `lilo -v` command.

```
[root@HostA /root]# lilo -v
LILO version 21.4-4, Copyright (C) 1992-1998 Werner Almesberger
'lba32' extensions Copyright (C) 1999,2000 John Coffman

Reading boot sector from /dev/hda
Merging with /boot/boot.b
Mapping message file /boot/message
Boot image: /boot/vmlinuz-2.4.3-12
Mapping RAM disk /boot/initrd-2.4.3-12
Added linux *
/boot/boot.0300 exists - no backup copy made.
Writing boot sector.
```

### Step 10. Last Check Before Rebooting

Just before you reboot, check to see whether the following commands now display the updated kernel information, and then rebuild the RPM database just to be safe. If the kernel version is correct, then you should log in and check the `uname` of the machine by using the `uname -a` command.

```
[root@HostA /root]# uname -a
Linux HostA 2.4.3-12 #1 Fri Jun 8 15:05:56 EDT 2001 i686 unknown
```

The `rpm -rebuilddb` command will resolve any RPM layout differences if there are any.

We recommend that you query the RPM database, check the system's disk space, create another updated boot disk, and so forth.

```
[root@HostA /root]# rpm --rebuilddb
```

### **Step 11. Reboot and Test the New Kernel**

```
[root@HostA /root]# reboot
Red Hat Linux release 7.1 (Seawolf)
Kernel 2.4.3-12 on an i686
login:
```

## **A Quick Guide for RedHat V6.X RPM Upgrade**

---

The exercise is straightforward. These instructions can be used on all versions of Red Hat Linux V6 as long as you ensure that the RPM package versions are consistent with the existing version of both the distribution and the kernel levels. This exercise is by example in the sense that the actual revision numbers on the RPM packages might vary from system level to system level. We encourage you to take good notes as you go along to ensure a successful upgrade.

1. Check to ensure that you have enough disk space for the kernel in `/boot` and for the source files in `/usr`.

Before:

```
# df
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/hda1 806368 55140 710264 7% /
/dev/hda5 272226 1551 256620 1% /home
/dev/hda7 964500 667672 247832 73% /usr
```

After:

```
# df
Filesystem 1k-blocks Used Available Use% Mounted on
/dev/hda1 806368 55140 710264 7% /
/dev/hda5 272226 1551 256620 1% /home
/dev/hda7 964500 667672 247832 73% /usr
```

2. Identify the RPM packages already installed on your system, and make a note of them. Compare them to the RPM packages that are required to upgrade the kernel.

```
# rpm -q --all | grep -i kernel
kernel-headers-2.2.14-5.0
kernel-pcmcia-cs-2.2.14-5.0
kernel-source-2.2.14-5.0
kernel-utils-2.2.14-5.0
```

3. Using FTP, download the files into a newly created directory. We downloaded our files from [www.redhat.com/support/errata/RHSA-2000-037-05.html](http://www.redhat.com/support/errata/RHSA-2000-037-05.html). If this site is too busy, go to [www.redhat.com/mirrors.html](http://www.redhat.com/mirrors.html) and try one of the alternate sites for download. After we downloaded the file, the file's integrity was checked.

```
# mkdir -p /usr/sysadmin/kernel
kernel-2.2.16-3.i686.rpm
[/root]# rpm -K --nopgp kern*
kernel-2.2.16-3.i686.rpm: md5 GPG OK
```

4. Compare the packages currently on the system to the downloaded packages that are to be installed onto the system. The following packages will be replaced:

```
# rpm -q --all | grep -i kernel
kernel-headers-2.2.14-5.0
kernel-pcmcia-cs-2.2.14-5.0
kernel-source-2.2.14-5.0
kernel-utils-2.2.14-5.0
```

The replacement RPM package is as follows:

```
# ls
kernel-2.2.16-3.i686.rpm
```

5. Before you upgrade the kernel, check and record the current version.

```
Red Hat Linux release 6.2 (Zoot) Kernel 2.2.14-5.0 on an i686
login:
```

6. This point is the last chance that you have to create an emergency boot floppy disk. If you have not already done so, make one now and label it as the boot disk.

```
# cat /etc/lilo.conf
boot=/dev/hda
map=/boot/
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.2.14-5.0
label=linux
root=/dev/hda1
read-only
# mkbootdisk --device /dev/fd0 2.2.14-5.0
```

Insert a disk in /dev/fd0. Any information on the disk will be lost. Press Enter to continue or Ctrl-C to abort.

7. Proceed to either upgrade or install the new RPM packages. You might get an error when you upgrade the kernel headers file due to the fact that it could not remove the old */lib/modules*. This situation is acceptable. Make a note to yourself to clean these up later.

```
# ls
kernel-2.2.16-3.i686.rpm
# rpm -Uvh kernel-2.2.16-3.i686.rpm
```

On other versions of the kernel, you might find that there are conflicts between packages. You might have to force the install. In this exercise, forcing the install is not necessary. Forcing the install would look like the following:

```
# rpm -ivh --force kernel-pcmcia-cs-2.2.5.22.i386.rpm
kernel-pcmcia-cs
```

8. If you have a SCSI system or a laptop that needs PCMCIA support, you might have to create an initial RAM disk. As we mentioned earlier, the purpose of this initial RAM disk is to enable a modular kernel, such as Linux 2.2.X, to have access to modules required for booting as the system boots up. Essentially, it copies the required module driver to the boot entity so that it is available.

```
# ls -l /boot total 2804
lrwxrwxrux 1 root System.map -> System.map-2.2.16-3
-rw-r--r-- 1 root System.map-2.2.16-3
-rw-r--r-- 1 root boot.0300
-rw-r--r-- 1 root boot.b
-rw-r--r-- 1 root chain.b
-rw-r--r-- 1 root initrd-2.2.16-3.img
-rw-r--r-- 1 root kernel.h
-rw----- 1 root map
lrwxrwxrwx 1 root module-info -> module-info-2.2.16-3
-rw-r--r-- 1 root module-info-2.2.16-3
-rw-r--r-- 1 root os2_d.b
-rwxr-xr-x 1 root vmlinux-2.2.16-3
lrwxrwxrwx 1 root vmlinux -> vmlinux-2.2.16-3
-rw-r--r-- 1 root vmlinuz-2.2.16-3
```

9. From this information, you can construct the `mkinitrd` command:

```
# mkinitrd /boot/initrd-2.2.16-3.img 2.2.16-3
```

10. Now, set up LILO to point to the new kernel:

```
# cat /etc/lilo.conf
boot=/dev/hda
```

```
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux
image=/boot/vmlinuz-2.2.16-3
label=linux
read-only
root=/dev/hda1
# lilo -v
LIL0 version 21, Copyright 1992-1998 Werner Almesberger Reading boot
sector from /dev/hda Merging with /boot/boot. Boot image:
/boot/vmlinuz-2.2.16-3 Added linux * /boot/boot.0300 exists - no
backup copy made. Writing boot sector.
```

11. Finally, shut down and test the new system. A very common issue is to lose the ability to use telnet, ftp, and other functions after an upgrade from V6.X to V7.X. If this happens do not worry, your configuration parameters are still in the */etc/rc* startup files. The upgrade process took the liberty of placing comment “#” symbols within the files to shut down those services. All you have to do is remove the comments inside these files, and the system will be back to normal.

```
# shutdown -r now
```

12. Log in again and check the name for the correct kernel version.

```
Red Hat Linux release 6.2 (Zoot)
Kernel 2.2.16-3 on an i686
login:
```

You should now make a new backup, but do not overwrite your existing backup for a while just in case something is not right. Test the system as best you can, and document any changes that you made and add them to your records.





# Configuring the Kernel

In this chapter, we discuss all of the available tools for configuring a Linux kernel. We then use `xconfig` to create an *xconfig* file used by the compile process. We show the necessary and optional configurable parameters for a Linux kernel based on hardware requirements. This chapter also makes various recommendations for handling major configuration issues for a more efficient kernel.

The only people who are interested in configuring their own kernels are developers, who are looking for code from the Internet that is newer than that offered by off-the-shelf Linux CD-ROMs, or system administrators, who know exactly what they want and are moving toward a very specific set of kernel components. This chapter is presented for two reasons. First, we will give those who have not had a chance to configure a kernel a process to start with. Second, we will illustrate how you can get details on what is available via Linux kernel functions by using the `xconfig` menus (many of these features would not be visible otherwise).

After discussing the Linux kernel configuration process, we move on to the default behavior of the most important kernel menus. Next, we identify



the menu options that invoke driver details, configure basic and crucial kernel options, and create a good *xconfig* file.

Before we start, we would like to say that you should not be intimidated by what you are about to see. Only about 20 percent of this material will apply to any particular machine at any one time. The material covered in this chapter is a fine demonstration of how sophisticated Linux has become over its short history, however. You will see support for almost everything in these menus. We will not define each item in the menus. At any time, however, you can open a menu and read the Help prompts. The more familiar you become with these menus, the easier Linux will be to learn. In addition, you should note that these choices are totally transparent if you choose to use the installation procedures for Linux shown earlier in this book, as opposed to configuring your own kernel.

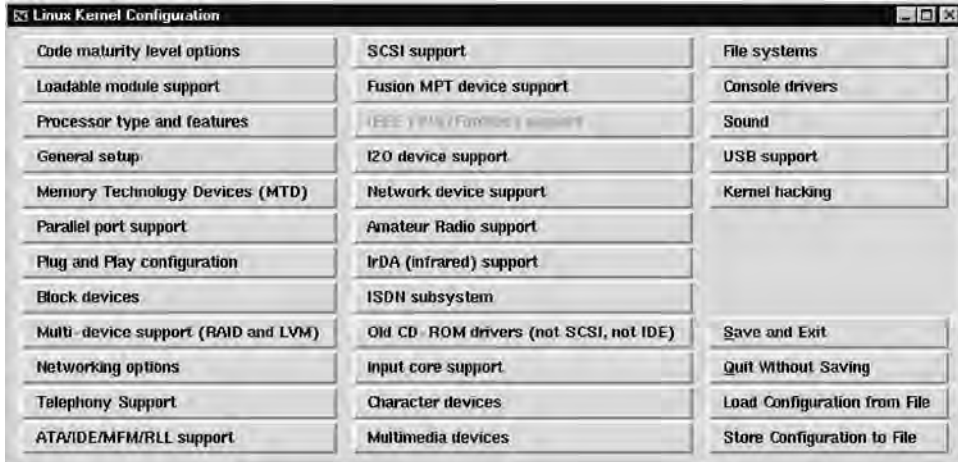
We find that many people are baffled by what is and is not supported in Linux. The easy answer to these kinds of questions is to access the `make xconfig` menu and simply look. No changes will be made to the system until you actually perform a compile, which has to be done by using `make bzImage`. You have to be positioned in the kernel config directory, `/usr/src/linux-2.4.3-12`, before `make xconfig` will execute. The text version of this tool is `make xconfig`. Although this utility is functionally equivalent, it is very difficult to handle. Given the choice, we recommend `make xconfig`.

## **Linux Kernel Configuration Menu**

---

The Linux Kernel Configuration start menu presented by `make xconfig` appears in Figure 9.1. We use this start menu as a guide only. We do not use every option or combination in the following discussion. We explain the mandatory options and make various recommendations. The general rules applied here are simply to compile infrequently used features as modules to reduce resource consumption and kernel size or to compile non-optional features such as disk-drivers and file-system support right into the kernel for performance reasons.

This menu looks daunting at first, but our advice is to visit only those features that you need for your specific system. View these menus in the same way as the package menus that are presented during the installation. Actually, these are a result of choosing those packages. These menus will dynamically change depending on the features that you enable. For example, if you choose not to enable SCSI devices, all options pertaining to SCSI devices will be dimmed and will thus be unavailable. For this reason, we



**Figure 9.1** The Linux Kernel Configuration start menu.

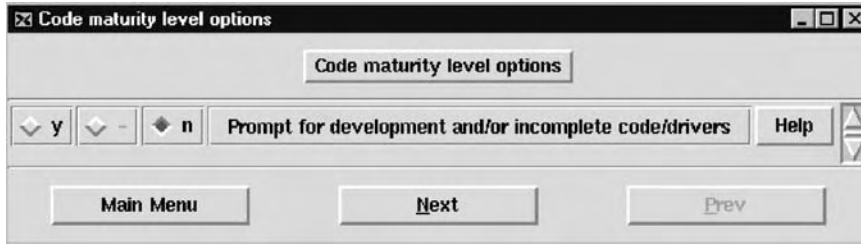
recommend that you work through these menus in an orderly fashion, from Code maturity level options to Kernel hacking.

As we have mentioned, for this menu and all subsequent menus in this chapter, we will not discuss every feature. We will cover only important points and issues that commonly cause problems. You will soon see that many of these features will not pertain to your kernel; hence, they can be ignored. Many are features that support hardware that you probably do not have inside your system. If you encounter a situation that requires information about a specific feature that we have not discussed in any detail, you can investigate it by simply using Help within the option.

## Code Maturity Level Options

The Code maturity level options (Figure 9.2) will decide the entire scope of what you are going to configure in this kernel. If you choose *y* in this dialog box, you will be allowing experimental code to be used—that is, code associated with the odd-numbered version of the kernel source tree. If you choose *n*, you will not allow the experimental code to be used. Note that some of the areas on the dialog box (for example, the - [dash] button, Prev, and the up and down arrows) are dimmed so they are not available for the option.

The *n* choice is recommended if you are working in a production environment; *y* is recommended if you are installing a development system. Whatever you choose here will determine what will be available to you in the subsequent menus. Therefore, if you are trying to install a new driver



**Figure 9.2** Code maturity level options.

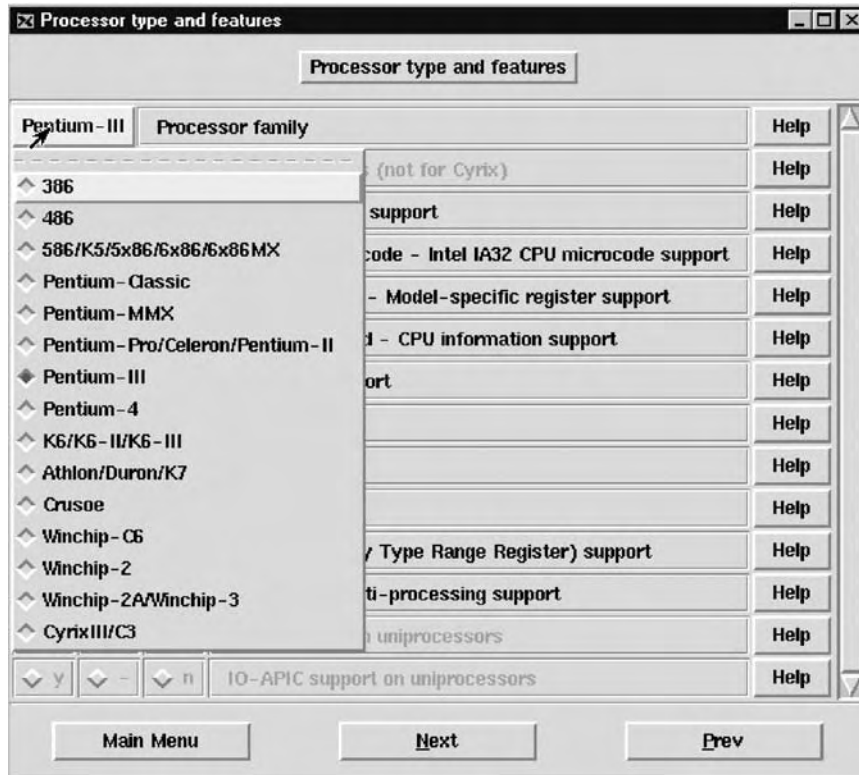
and it is dimmed and seemingly unavailable, return to this Code maturity level option. In addition, change it from `n` to `y`. Now, the otherwise unavailable driver will likely become available.

Making selections in this dialog box should be part of a larger strategy to determine which kernel version you have installed on the system. For example, all critical systems should be running kernels whose version numbers are even numbers. We discuss kernel version numbers in Chapter 10, “Installing a New Kernel by Building a New Source Tree.” Using experimental code—that is, kernels whose version numbers are odd numbers—is not recommended in critical production systems, either. You should not choose the experimental code feature when using an even-numbered kernel source unless you have an overwhelming reason to do so.

## Processor Types and Features

For performance reasons, it is recommended that you choose the correct processor for the system. If you do not know what processor you have, you can use 386, which is the default. It will work on newer systems, but it might not perform as well. The processor family used to generate the kernel that is already installed on the system is displayed as part of the login herald or can be displayed by using the `uname -a` command.

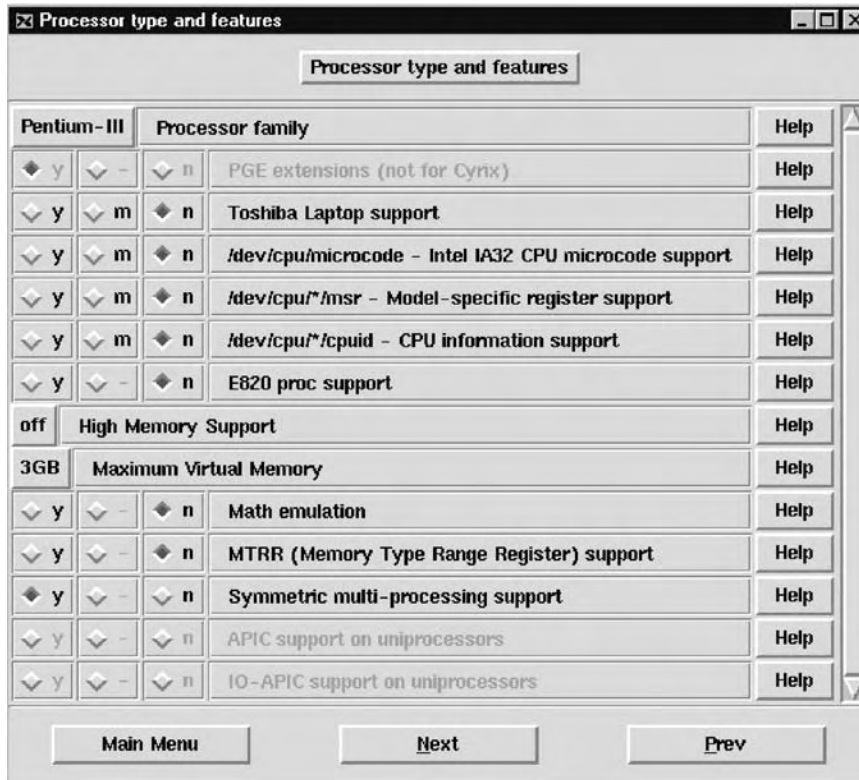
Choose the Math emulation option shown in Figure 9.4 only if your system does not have a math coprocessor chip. It adds a whopping 45KB of overhead to your kernel. All Pentium-class machines have the coprocessor function built in; therefore, you should not have to choose this option. This option is not asking whether you have math emulation. Rather, it is asking, “If you don’t have it, do you want a software equivalent?” The only systems that require this option are the older i386DX, i386SX, or i486SX systems.



**Figure 9.3** Processor family.

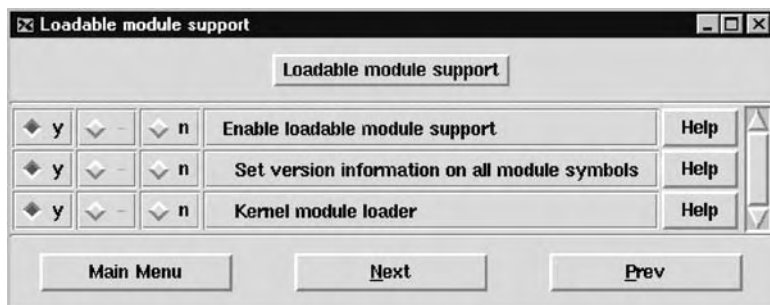
## Loadable Module Support

Choosing Enable loadable module support (Figure 9.5) represents a kernel strategy and will determine whether you install a “monolithic” kernel or a “modular” kernel. One of the most important features of the Linux 2.X kernel is its capability to be modular. According to the modular kernel concept, certain options need only be loaded “on demand” (that is, only when they are required). When those options are not required, they are unloaded. Thus, the size of the kernel, which is loaded in RAM, is kept to a minimum—and resources are not tied up any longer than necessary. For performance purposes, however, we recommend that any feature that you expect to use regularly should be non-modular. By choosing even one module, however, you are committing to a modular kernel. Conversely, if you choose no here, you will find that some features might not be available to you.



**Figure 9.4** Processor type and features.

The Set version information on all symbols for modules option means that you can use modules from other kernel versions as long as they are compatible.



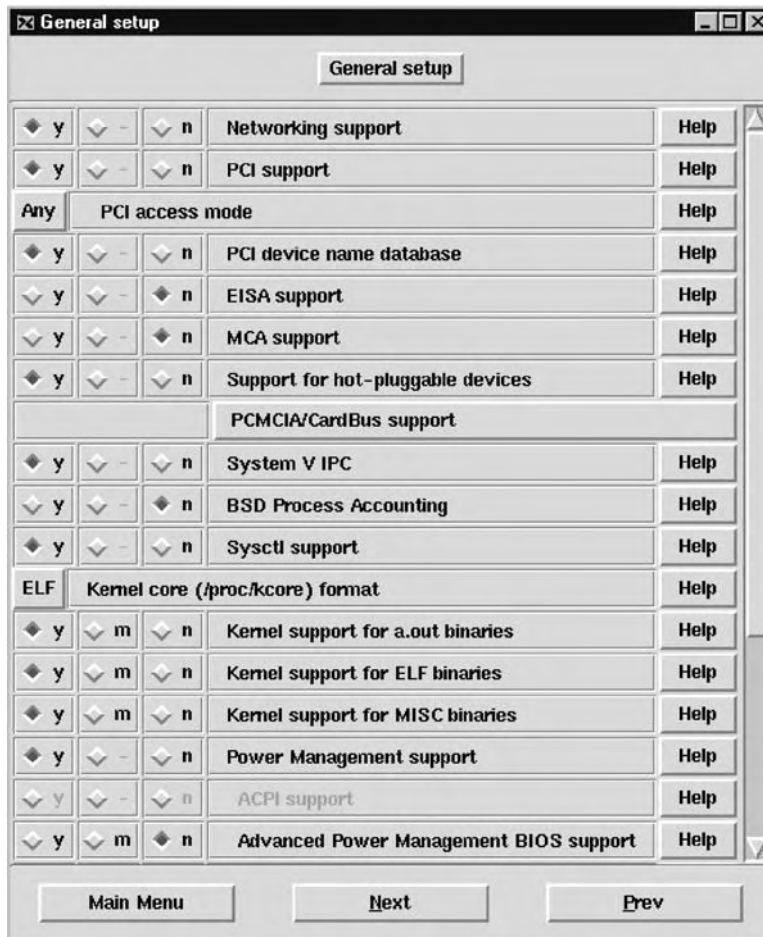
**Figure 9.5** Loadable module support.

The final option in this dialog box, Kernel module loader, is the `kernelld` daemon support. The `kernelld` is the daemon that actually performs the automated loading and unloading of modules.

We recommend that you choose `y` for all three options in the Loadable module support dialog box.

## General Setup

The General setup dialog box has many options, so we present a brief look in Figure 9.6 and talk only about the most important issues.



**Figure 9.6** General setup.

### ***Networking Support Option***

If you are going to have a network adapter in your system, you should answer *y* to the Networking support option (Figure 9.6). Answering *n* will dim the network-adapter drivers in all subsequent menus.

### ***PCI Support Option***

PCI support (Figure 9.6) refers to the type of bus that you have. If you have a PCI motherboard, then answer *y*; otherwise, answer *n*. In other words, answer *n* if your bus is ISA, EISA (Extended Industry Standard Architecture), MCA (Microchannel Architecture), or VESA. If you have an ISA bus with a couple of PCI slots, answer *n*.

### ***PCI Access Mode Option***

Linux can try to determine the type of PCI devices in your system via the BIOS by querying the device directly or by trying other strategies. Note that the PCI access mode option (Figure 9.6) does not require the PCI support option to be selected. The PCI device can actually be inside an ISA system in a PCI slot.

### ***PCI Quirks Option***

The PCI quirks selection provides backward-compatible code for PCI environments with broken or obsolete BIOSs.

### ***System V IPC Option***

Indicate *y* for the System V IPC option (Figure 9.6) if you want DOS environments to work properly.

### ***Kernel Core Option***

It is highly recommended that you enable all three Kernel support options (for ELF, MISC, and JAVA binaries) shown in Figure 9.6 for compatibility purposes. You should enable them particularly if you have enabled Prompt for development and/or incomplete code/drivers back in the Code maturity level options dialog box (Figure 9.2).

### ***Parallel Port Support Option***

If you are going to use the parallel port for a printer or other peripheral devices, then say *y* to the Parallel port support option inside the General Setup. This option is one of those options that if set to *n* now, it will likely cause problems later.

### ***Advanced Power Management BIOS Support Option***

The Advanced Power Management BIOS support option in Figure 9.6 is useful for laptops. It has nothing to do with options such as disk drive spin-downs or green monitors, however. Besides, those features are not governed by operating systems but rather are built into the devices themselves.

### **Plug and Play Support Option**

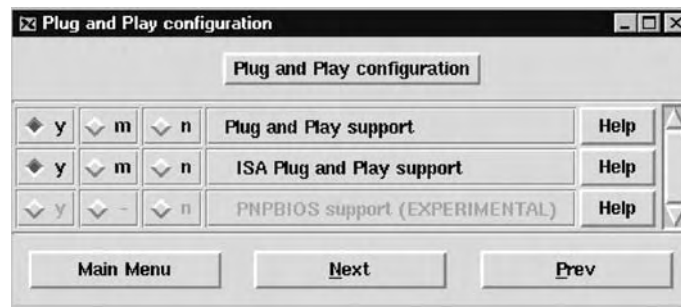
You will likely benefit from enabling the Plug and Play support, especially to facilitate probing. You enable it through the dialog box shown in Figure 9.7.

### **Block Devices**

The Block devices option requires many subsequent choices. We present these choices in Figures 9.8 and 9.9.

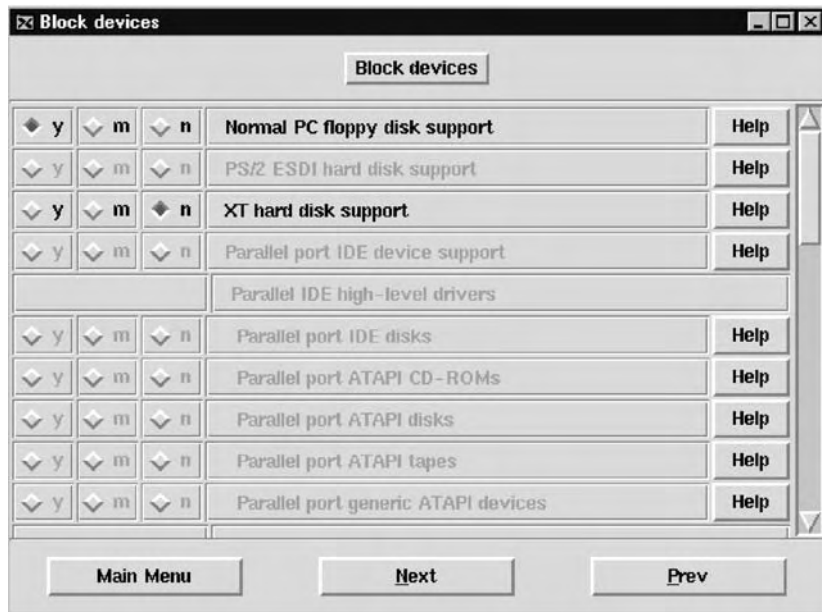
### ***Normal PC Floppy Disk Support Option***

If you want to use your floppy disk drives with Linux, you have to select the Normal PC Floppy disk support option shown in Figure 9.8.

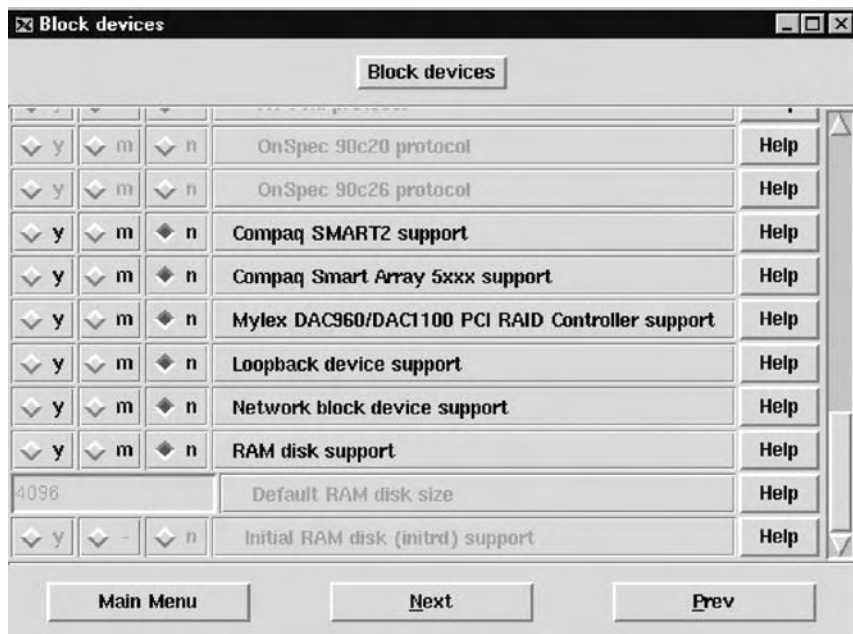


**Figure 9.7** Plug and Play configuration.





**Figure 9.8** Block devices: bottom part of menu.



**Figure 9.9** Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support option.

If you have any type of IDE drive in your system, select `y` for the Enhanced IDE option. This driver is preferred over the one listed a few features below it. Include IDE/ATA-2 DISK support. If the system is 100 percent SCSI, you can select `n`. You could also enter `m` here; if you need this type of support later, you can load it as a module.

### ***Include IDE/ATA-2 DISK Support Option***

Originally, IDE technology supported two drives: one was the master and one was the slave. Multiple IDE drive support came later. The Include IDE/ATA-2 DISK support option applies to an older version of an IDE driver that might be required for systems with dated IDE technology. If you have an up-to-date system, try the Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support first, and only try this option if the Enhanced option does not work.

### ***Include IDE/ATAPI Options***

For the Include IDE/ATAPI CDROM support, Include IDE/ATAPI TAPE support, and Include IDE/ATAPI FLOPPY support options, we offer the same comment as earlier.

### ***SCSI Emulation Support Option***

Many new devices on the market today, such as CD-ROM writers and certain backup devices, are IDE devices—but they emulate a SCSI device. We recommend that you choose `m` for the SCSI emulation support if you have such devices unless kernel size is an issue.

### ***Other IDE Chipset Support Option***

If you are having a lot of trouble getting the older IDE drives to be recognized, try enabling Other IDE chipset support. In some cases, it can make a difference.

Some of these support options are for odd stuff. Someday, when you have nothing else to do, you can become familiar with these by browsing all the Help option panels. We recommend that you just make a mental note of the options here.

### ***Loopback Device Support Option***

Activating the Loopback device support option (Figure 9.9) is required for the Common Desktop Environment. Saying *y* here will also enable you to use a regular file as a block device. This feature is useful if you want to check an ISO 9660 file system before burning a CD or if you want to use floppy images without first writing them to a floppy, as well as many other things. We recommend that you choose *y* or *m* here.

### ***Network Block Device Support Option***

The Network block device support option in Figure 9.9 is not your network adapter and driver information. Network adapter and driver support is in the next section.

### ***Multiple Devices Driver Support Option***

If you plan on bridging, striping, or mirroring disk sets, you had better select *y* for Multiple devices driver support.

### ***RAM Disk Support Option***

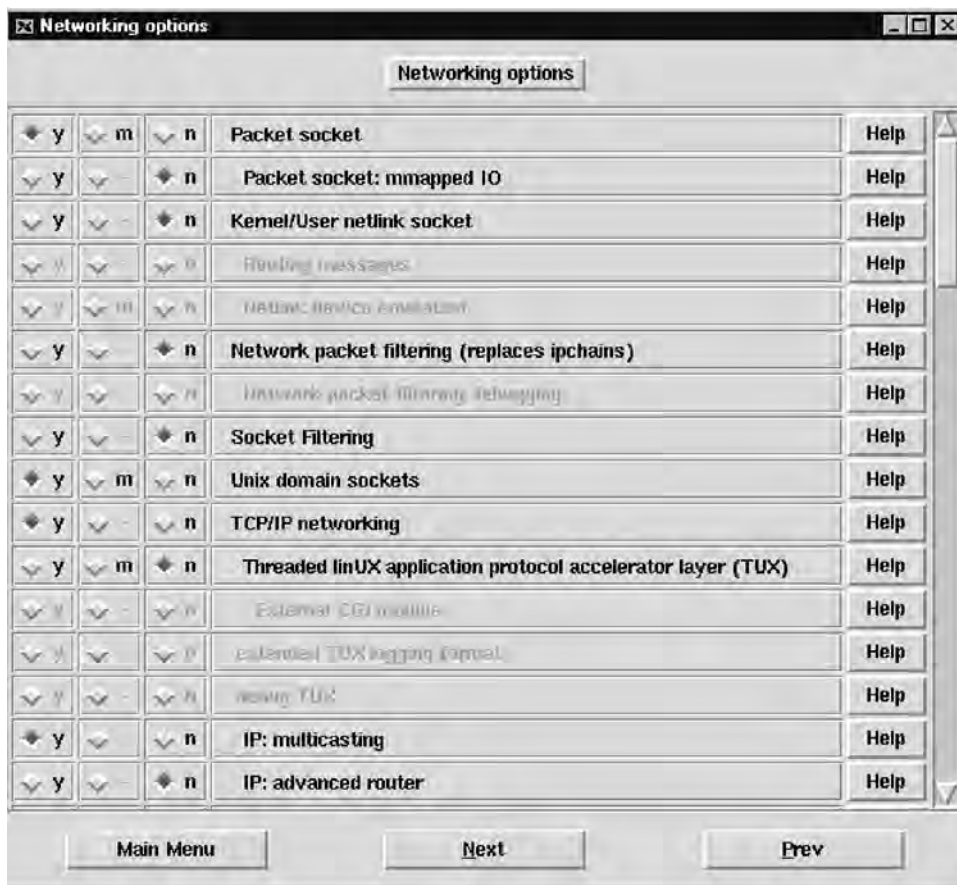
If you have either a SCSI device or a laptop requiring PCMCIA support, we recommend that you answer *y* to the RAM disk support option (Figure 9.9). Otherwise, you will not be able to create a RAM disk file by using the `mkinitrd` command to accommodate booting the system with preboot module requirements, such as SCSI drivers or PCMCIA support.

The SCSI support option is discussed a bit later. If you intend to install SCSI support then, you had better select RAM disk support now.

Here, you find relatively esoteric support programs and protocols. Just have a look at them and move on. If you are curious, check out their respective Help buttons.

## **Networking Options**

It is a good idea to have a clear strategy or knowledge of which networking features will be required and installed on your system. Arbitrarily enabling all options is inappropriate and insecure. We are not going to delve into definitions of TCP/IP now. It is assumed, however, that your system will participate as a typical host on a network.



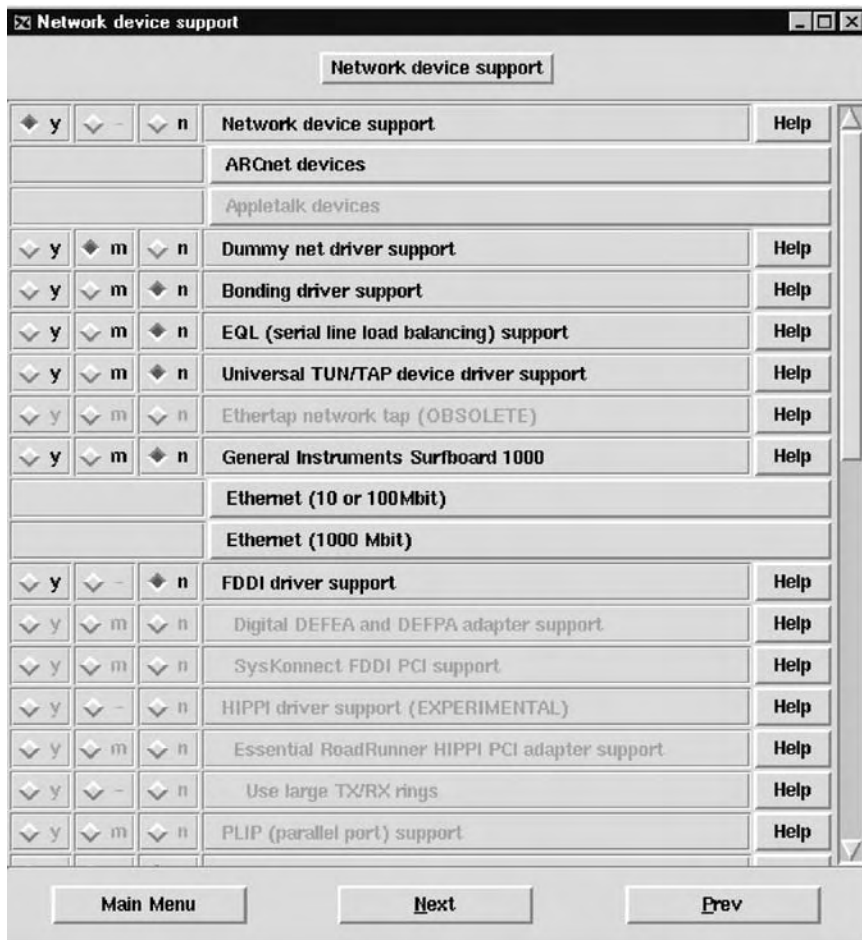
**Figure 9.10** Networking options.

### ***UNIX Domain Sockets Option***

The UNIX domain sockets option (Figure 9.10) is required for X protocol and daemon communication support. We recommend that you answer y here to enable basic TCP/IP support.

### ***TCP/IP Networking Option***

Select TCP/IP networking (Figure 9.10) to enable all the basic IPv4 TCP/IP protocols.



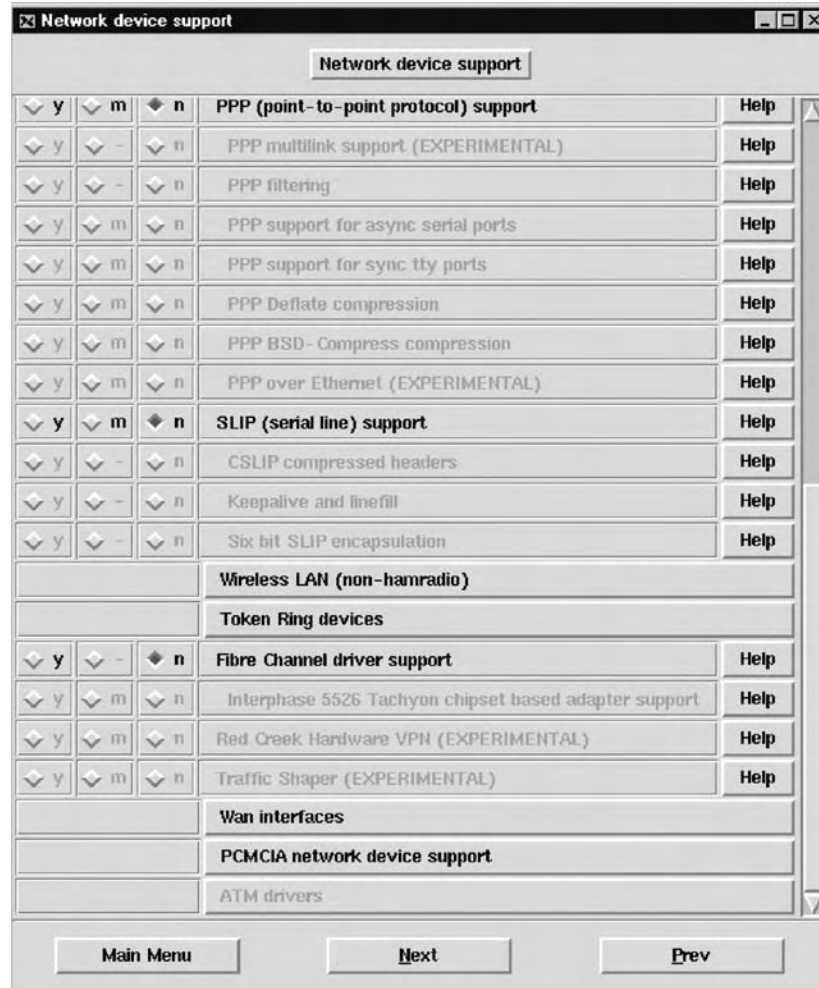
**Figure 9.11** Network device support: top part of menu.

## Network Device Support

Network device support is another large group of options. We present them in Figures 9.11 and 9.12.

### *Network Device Support Option*

Select the Network device support feature (Figure 9.11) if you have an adapter card to configure. Answering n will dim all possibilities for network adapter choices.



**Figure 9.12** Network device support: bottom part of menu.

### ***Dummy Net Driver Support Option***

The Dummy net driver support option (Figure 9.11) is useful if you are going to be enabling either SLIP (serial-line IP) or PPP (point-to-point protocol) support. Neither of these protocols will work if you do not answer either **y** or **m** here.

### ***Ethernet (10 or 100Mbit) Option***

Select Ethernet (10 or 100 Mbit) option (Figure 9.13) if you have an Ethernet adapter in your system. Then, proceed to locate the driver that matches



**Figure 9.13** Ethernet (10 or 100 Mbit option).

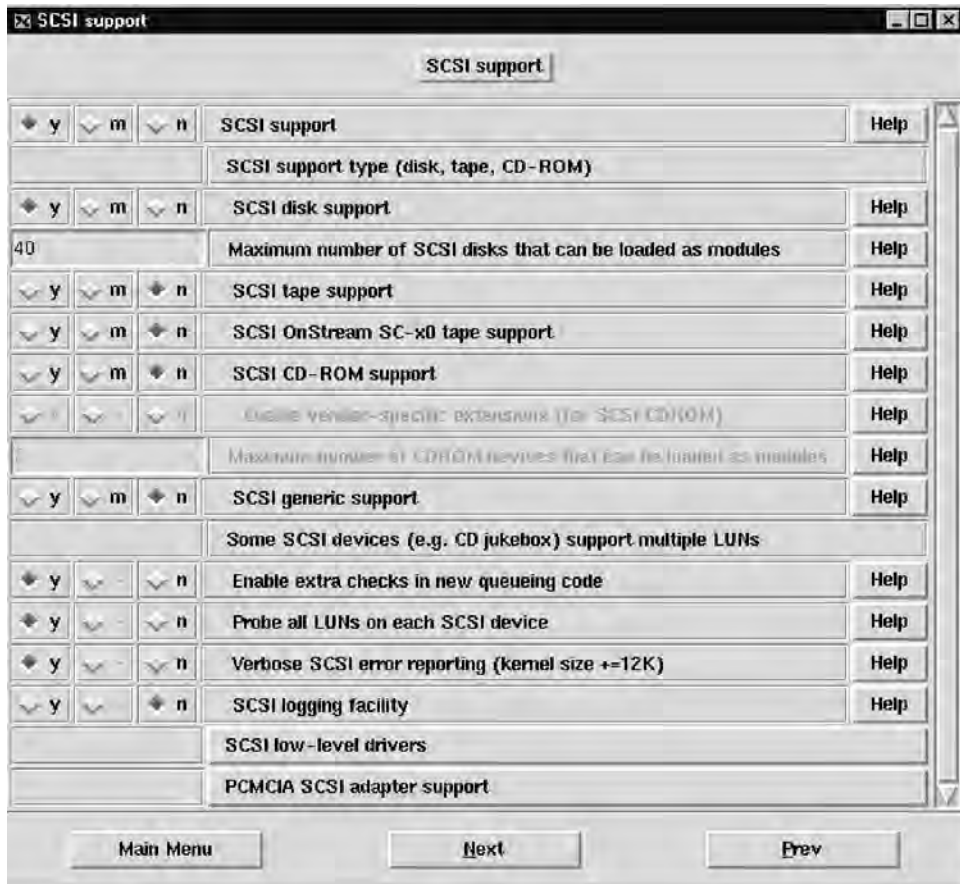
your adapter. If your driver does not show up here, you might have to return to your source tree and add a patch file to include your driver.

At this point, you can continue down the list and choose either *y* or *m* to indicate the drivers that you wish to have compiled into the system.

These are not common network devices, but they do exist. You might encounter them in your travels (or perhaps "travails" would be more appropriate).

## SCSI Support

If you are planning to use SCSI devices on your system, then you should select *y* or *m* for the SCSI support option in Figure 9.14, where *y* is the preferred selection.



**Figure 9.14** SCSI support.

If you choose SCSI support, then you have to be sure that you selected RAM disk support earlier (see the “Block Devices” section). Doing so ensures that the necessary SCSI modules and utilities will be loaded at boot time so that patches and upgrades can be applied if and when necessary or desired.

### ***SCSI Low-Level Drivers***

If you chose y for SCSI support in the previous section, then you must advise the installation program of the type of SCSI adapter or bus installed on your system. Choose the appropriate equipment from the list. A partial list is shown in Figure 9.15.





**Figure 9.15** SCSI low-level drivers.



**Figure 9.16** Filesystems.



**Figure 9.17** Kernel build instructions.

## Filesystems

If you choose to add NTFS file system support (read only) to the kernel (see Figure 9.16)—whether monolithic or modular—you will be able to access files on a Windows NTFS file system while Linux is up and running.

## Save and Exit

Save and exit! This advice might seem obvious, but many people do forget this step. When you are at this point, this entire procedure should have created an xconfig file that will be used to compile the kernel. See Figure 9.17.

We would like to make one final point before moving on to the next chapter. If later you have trouble running a command, configuring a port, and so on, ask yourself whether there is a configurable parameter missing from the compiled kernel. By now, you realize that files on your system are no guarantee of support in the kernel.



## Installing a New Kernel by Building a New Source Tree

Chapter 8 focused on installing a new Linux kernel by installing RPM packages, and Chapter 9 covered the basics of configuring a new kernel. In this chapter, the focus is on installing a new kernel beginning from the very basics—beginning with downloading the kernel in tar file format from a source Web site on the Internet. Then, we create all the new directories and symbolic links to prepare for the new source tree and configure, compile, and implement the new kernel.

You might have to modify steps presented in this chapter based on the particular version of code that you are planning to use. This job is probably the most complex task that you will ever have to perform on Linux, and it requires some knowledge of basic system administration (such as creating new directories and symbolic links). We do not assume that you can do this job without assistance, however; thus, we provide examples of syntax throughout the chapter.

You might have read about a utility called `zli10`. This utility attempts to configure LILO to boot your new kernel, and although theoretically useful, it can fail the test for multiple kernels (which is what we are trying to achieve).

We will assume that you have an existing functional system, and we are going to have you preserve your existing kernel configuration file by using the `make oldconfig` method. This procedure will enable you to get through this exercise without committing mental suicide. Once you have mastered the many kernel configuration options, we suggest using `make xconfig`.

As we proceed, we are going to encourage you to occasionally use modules as opposed to compiling an option right into the kernel. The results will show up when you test your new system after it is compiled. For example, if you choose to make your networking modular, you have to use a tool such as Red Hat, System, Control Panel to access your Ethernet interface to actually activate it or read the man pages on `insmod`. If nothing else, it will illustrate both the positive and negative aspects of using modules. If this process becomes very confusing to you, you can simply go back and change your configuration to compile the feature into the kernel as opposed to being modular. Having said that, however, if you use the `make oldconfig` method and use your previous `.config` file, then the new kernel should behave similarly to the old kernel. Starting with an existing `.config` file will reduce the possibility that the new kernel will not work due to a missing kernel prerequisite. This situation will enable you to focus on the implementation aspect of getting an updated kernel with the same features and functionality as before.

## Modules

---

We would like to talk about kernel modules from the point of view of a system administrator. There are the developers who create them and the system administrators who use them. The question is, what are they—and how are they used?

### **The Modules Concept: How They Work**

All early Linux kernels used to be monolithic. Monolithic means that all the code and functionality that you needed was compiled into a single kernel file. This feature is very efficient if you are constantly using all the resources all of the time. On the other hand, if there is a resource that is used only occasionally, then it is not as efficient because you have to have that code loaded all the time—even when the resource is not being used.

The concept of the module is to have a compiled code segment that can be loaded and unloaded as desired. The modules can be loaded manually by using the `insmod` command or can be coded to be dynamic. Dynamic implies that the kernel will load the modules when a call for that resource is made. The trick here, of course, is performance, whereby there might be a slight delay while the kernel loads the module before the resource can be accessed. This situation is referred to as the *life cycle*. The life cycle behavior is coded into the module by the programmer and hopefully is documented.

An example would be the modules that are loaded at boot time. During the boot, the kernel will make a call for a module. These are the code elements that by virtue are modules or were chosen to be modules during the `xconfig` process. The kernel will call `kernel_d` to load the appropriate module. The `kernel_d` will call either `modprobe` or `insmod` to actually load the module. These utilities, in turn, use the `sys_create_module` to allocate memory resources and `get_kernel_syms` to access the kernel symbol table for linking, and then call `init_module` to initialize the module.

Of course, the only reason to invoke these features is to be able to unload the module when you are done. The `rmmmod` unloads the module and also checks to see whether any other module or process is not currently using it. If it is in use, it will respond accordingly. Those modules that were loaded by `kernel_d` will be unloaded by `kernel_d` when they are no longer used. There is an idle timeout feature that expires that triggers a system call for the expired modules to be flagged as autoclean. Anything marked autoclean is unloaded, and all resources are released back to the system.

Before you can effectively use a specific module, you have to find out what the behavior of the modules is. Only the documentation for the module can tell you what you need to know. You can make assumptions about the behavior, but you might be wrong. For example, you might be trying to recompile or patch a kernel, and during the process you ask for your networking device drivers to be modular. After the kernel is compiled, the networking modules refuse to load or you get symbol errors. The reason is probably because although the modular option was available, the driver code is not compatible. Or, during the upgrade or patching procedure the module files might have become corrupt or were not of the correct version level. There are many variables.

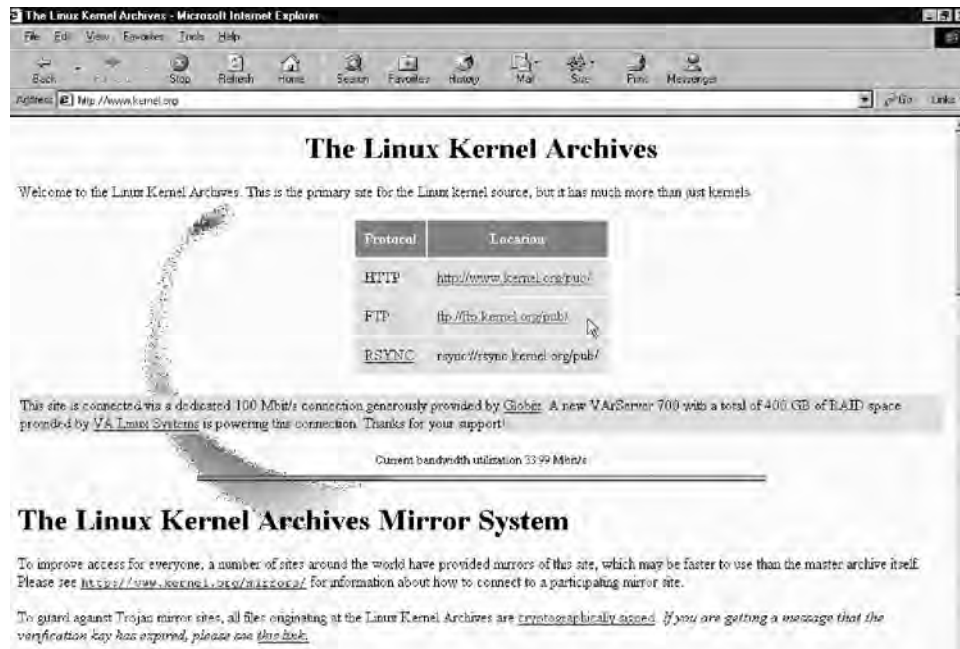
Modules are going to be the future, and there is a large push to code modules. Unfortunately, information and documentation on these modules is still somewhat difficult to come by. Information on exactly how to handle module issues during upgrades to properly use the modules is still not available.

The kernel is the core of the operating system, which manages and controls memory and manages processes and hardware access. Technically, users do not communicate directly with the kernel. The user interfaces with the kernel via a shell of some kind, like the bash shell. Modules interface with the kernel via an exported symbol set. These symbol sets give the modules access to kernel structures and routines. The symbol sets are a strict communication mechanism that will enable modules to share kernel resources in a similar sense to that of many users who are using the bash shell. See Figure 10.1.

Most kernel modules at the current time include device drivers for networking interfaces and file systems. They can be dynamically linked to the kernel at boot time via a script or manually by using any of the module utilities.

## Module Utilities

**depmod.** Calculates the dependencies between modules and the kernel and between one module and another. For example, module `nfsd`



**Figure 10.1** Linux kernel modules.

needs modules `lockd` and `sunrpc` to be loaded first. All three modules need certain symbols to be exported by the kernel before they will load. `depmod` is typically run once at boot time via the `/etc/rc.d` file. You can also run it before booting a new kernel for the first time or before storing a kernel on read-only media to make sure that what you are backing up is current.

**modprobe.** The high-level module loader. It will automatically extract and apply any module options that you specified. It also automatically loads any required modules (for example, when loading `nfsd`, `modprobe` will load `lockd` and `sunrpc` first). `modprobe` also has a mode where it automatically removes modules, again following the dependency chain. The idea is that `modprobe` will look first in the directory containing modules compiled for the current release of the kernel. If the module is not found there, `modprobe` will look in the directory common to the kernel version (2.4.2-2 or 2.2.17-14). If the module is still found, `modprobe` will look in the directory containing modules for a default release (and so on).

**insmod.** The low-level module loader. It does exactly what you tell it to do, no more and no less. If you want any options, you must specify them on the command line. You have to load any required modules first, however, because `insmod` only loads one module at a time.

**rmmod.** Removes a module. Modules can only be unloaded if they are not in use. Being in use could be caused by a file being open on a file system, a network interface being up on a driver, or a module being marked as in use because another module depends on it.

**lsmod.** Lists modules that are loaded and their status, including dependencies.

**kmod.** This *is* actually part of the kernel. This feature is optional, and when compiled into the kernel, it will automatically load modules “as required.” An earlier version of `kmod` was called `kernelld`, but that has long since died.

## Main Modutils Files

`/lib/modules/2.4.2-2/` is the default directory containing modules for kernel 2.4.2-2. Modules can be loaded from anywhere, but the default is `/lib/modules/'uname -r'/`. Each time you compile a new kernel, the command `make modules_install` will create a new directory but will not change the default. For this reason, after performing an upgrade you have a new



set of modules, but the links are all still pointing to the older modules. This feature is a default strategy that can be changed in `/etc/modules.conf`.

When you install a new Linux, the modules should be moved to a directory related to the release (and version) of the kernel you are installing. Then, you should do a symlink from this directory to the default directory.

When you acquire a module that is unrelated to the kernel distribution, you should place it in one of the version-independent directories under `/lib/modules` to keep it separate for the purpose of upgrades and patching.

- `/lib/modules/2.4.2-2/modules.dep` was built by `depmod` and contains the inter-module dependencies.
- `/etc/modules.conf` is the configuration file for `modprobe`. It contains options for individual modules, aliases to map generic names to specific drivers, additional inter-module dependencies, and so on. See the `man modules.conf` for details.

```
[root@HostA /etc]# more modules.conf
alias eth0 ne2k-pci
alias parport_lowlevel parport_pc
alias usb-controller usb-uhci
```

## ***Upgrade Issues with Modules***

Most issues evolve around two basic issues:

- It is not well understood that after an upgrade or recompile, new modules are made but are not linked.
- It is up to the system administrator to copy the module files and symlink them.

The second issue has to do with the version dependencies between the modules themselves and the kernel symbol tables, particularly with those modules sourced from independent code producers or those modules distributed in binary form. The reason why this situation is problematic is because the modules should be recompiled for each kernel version. And then, the modules are compatible with the new symbol tables that were supplied with the new kernel. They say that the module becomes incompatible with a different kernel version when the interface symbol table set offered by the kernel is modified. This situation will result in the following error message:

```
Kernel does not load module - Unresolved symbol error messages
```

When you boot the kernel and the system tries to load any modules and you get “Unresolved symbol : \_\_some\_function\_name,” then it means that you did not clean compile the modules and kernel.

The solution is to recompile from make clean:

```
# cd /usr/src/linux
# make dep
# make clean
# make bzImage
# make modules
# make modules_install
```

### Kernel Fails to Load a Module

If the kernel fails to load a module (say, a loadable module for a network card or other devices), then you might want to try to build the driver for the device right into the kernel. Sometimes the loadable module will *not* work and the driver needs to be built right inside the kernel. For example, some network cards do not support the loadable module feature—you *must* build the driver of the network card right into the Linux kernel.

The solution is to revisit xconfig and reset that code not to be modular and try again.

### Reinstall Loadable Modules

If you are really desperate, you can reinstall the loadable modules. There is a chance that the files inside the `/lib/modules/2.4.2-2` are damaged or are an incorrect version to that of the kernel. In previous versions of Linux, the modules were in a separate rpm called kernel-modules, but now they are in the kernel rpm file. So, to replace the modules, you have to force reinstall the kernel rpm. Be sure that you install the correct CPU architecture. We do not recommend that you perform this task before a backup.

```
# rpm -qa | grep -i kernel

# rpm -U --force /mnt/cdrom/Redhat/RPMS/kernel-2.4.2-2.i686.rpm
(or)
# rpm -U --force /mnt/cdrom/Redhat/RPMS/kernel-2.4.2-2.i586.rpm
(or)
# rpm -U --force /mnt/cdrom/Redhat/RPMS/kernel-2.4.2-2.i386.rpm
```

Earlier Versions can use this syntax:

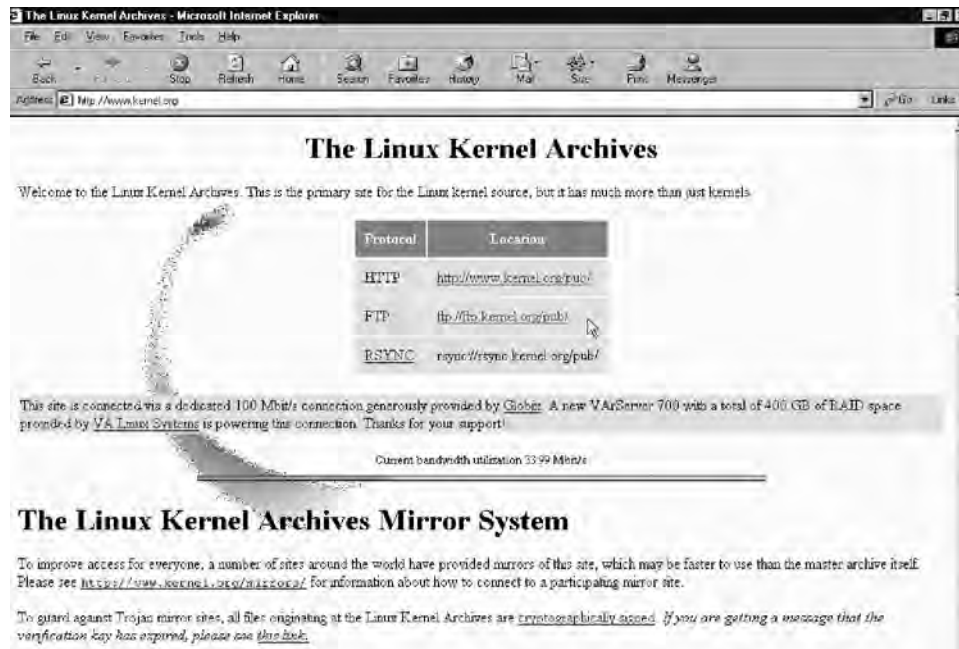
```
# rpm -i /mnt/cdrom/contrib/kernel-modules*.rpm
```

## Finding a New Kernel

To install a kernel from scratch, we will start by obtaining a tar file from a reputable Web site and then creating the necessary file structures to recompile the kernel and rebuild the system. The most popular Web site to begin from is the Linux public archive at [www.kernel.org](http://www.kernel.org), which is shown in Figure 10.2. As you can see from the figure, however, there are other trusted “mirror” sites from which the kernel files can be downloaded, which might be faster.

## Installing a New Kernel

Installing a new kernel by using a source tree requires more expertise than using RPM packages. There are more than a dozen tasks to complete. These steps are not always the same on every system. We take you through this process over the course of the next 18 steps. Performing these tasks will also demonstrate how the operating system is assembled and will introduce its internal operations.



**Figure 10.2** Linux public archive and mirror sites.

### Step 1. Record Current Kernel Level and Create Boot and Rescue Diskettes

The first component of this step, recording kernel information, is important because there is some information here that you will need to help you decide how to handle some of the compilation parameters. Repeating the procedure at the end of the process and comparing the two will indicate that the files were indeed properly upgraded or handled.

In the following example command lines, the command `uname` tells Linux to display system information. The `-a` option tells Linux to display all of the following information: hardware type, network-node host name, operating system release, operating system name, and operating system version.

```
[root@HostA /root]#uname -a
Linux HostA 2.4.2-2 #1 Sun Apr 8 20:41:30 EDT 2001 i686 unknown
```

We recommend recording the output from the `uname -a` command. Record the exact information on the kernel version and the architecture, which in the example case here is `i686`. An alternative way to check the version information is to `cat /proc/version`.

If you have not already done so, create a boot and rescue disk set now. In a production environment, you should ensure that you have a complete backup, as well.

```
# mkbootdisk --device /dev/fd0 2.4.2-2
Insert a disk in /dev/fd0. Any information on the disk will be lost.
Press <Enter> to continue or ^C to abort:
```

If the existing version of Linux is older than Linux 6.1, you might also need to use the following two commands to pre-mount the CD-ROM:

```
# mount /mnt/cdrom
# dd if=/mnt/cdrom/images/rescue.img of=/dev/fd0 bs=1440k
```

We also recommend, purely for academic reasons, that you document the links and directories in the boot directory before you start—because this information will change after the new kernel is installed.

```
# cd /boot
# ls -l
total 3484
-rw-r--r-- 1 root root 512 Jul 3 13:27 boot.0300
-rw-r--r-- 1 root root 5824 Feb 23 12:33 boot.b
```

```

-rw-r--r--      1 root    root      612 Feb 23 12:33 chain.b
lrwxrwxrwx      1 root    root      14 Jul  3 13:22 kernel.h ->
kernel.h-2.4.2
-rw-r--r--      1 root    root      405 Jul  3 13:23 kernel.h-2.4.2
drwxr-xr-x      2 root    root     12288 Jul  3 13:19 lost+found
-rw-----      1 root    root     12288 Jul  3 13:27 map
-rw-r--r--      1 root    root    23108 Feb 23 12:33 message
lrwxrwxrwx      1 root    root      19 Jul  3 13:22 module-info ->
module-info-2.4.2-2
-rw-r--r--      1 root    root    13598 Apr  8 18:57 module-info-2.4.2-2
-rw-r--r--      1 root    root      640 Feb 23 12:33 os2_d.b
lrwxrwxrwx      1 root    root      18 Jul  4 10:43 System.map ->
System.map-2.4.2-2
-rw-r--r--      1 root    root   410990 Apr  8 18:57 System.map-2.4.2-2
-rwxr-xr-x      1 root    root   2280836 Apr  8 18:57 vmlinux-2.4.2-2
lrwxrwxrwx      1 root    root      15 Jul  3 13:22 vmlinuz ->
vmlinuz-2.4.2-2
-rw-r--r--      1 root    root   781806 Apr  8 18:57 vmlinuz-2.4.2-2

```

## Step 2. Obtain the Source Code in tar File Format

Figure 10.3 illustrates a listing that might be found at a typical Linux kernel source mirror site. Note that these tar files have a very specific naming convention that provides information on the state of the Linux kernel. For example, a kernel source tar file that has a minor version number that is an odd number is deemed a development release and might contain bugs. On the other hand, a tar file whose minor version number is an even number is considered a better, more stable release.

## Comments

For comments about the web pages, please send mail to [webmaster@kernel.org](mailto:webmaster@kernel.org).

For comments about the file archive, please send mail to [ftpadmin@kernel.org](mailto:ftpadmin@kernel.org).

Please do not send general Linux questions to these addresses. We do not have the resources to reply to them.

The latest stable version of the Linux kernel is: **2.4.5** 2001-05-26 01:26 UTC [Changelog](#)  
 The latest prepatch (alpha) version appears to be: **2.4.6-pre6** 2001-06-28 00:23 UTC [Changelog](#)



Linux is a Registered Trademark of Linus Torvalds  
[\(Legal Notice\)](#)

**Figure 10.3** Typical listing at a Linux kernel source mirror Web site.

Let's take a closer look at the numbering convention. The format is major-version number, minor-version number, revision number, and tar file type. The major-version number rarely changes. We have been using the same major version of the kernel for several months now. The minor-version number is an indication of stability. As we mentioned earlier, if the number is even, then the kernel is a more stable "production" version of the kernel. Odd-numbered minor numbers indicate that the kernel is still under development. This minor-version-number release can cause problems for novice users. Development kernels are necessary for developers. The revision number indicates the patch level of the respective kernel. These can change frequently. See Figure 10.4.

Red Hat is very particular about deeming a version to be production quality. At this writing, the following is stated on [www.kernel.org](http://www.kernel.org): the latest stable version of Linux is 2.4.4; the latest beta version of Linux is 2.4.5; and the latest prepatch (alpha) version appears to be 2.4.6-pre6.

The files will be in .tar, .tar.gz, or .tar.bz2 format. The gz and bz2 files are files compressed with gzip and bzip2, respectively, which means that you need the gzip and bzip2 utilities to unzip them.

Put the tar file into the `/usr/src` directory:

```
[root@HostA src]# ls /usr/src
core  linux-2.4  linux-2.4.2  linux-2.4.5.tar.gz  redhat
```

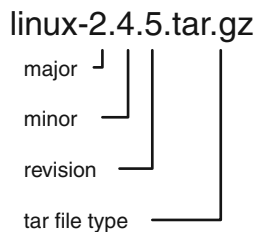
Do not unpack the files yet, however. At this point, simply note that when you unpack these files, you have to use the appropriate syntax.

For the .tar.gz files, use `tar xzpvf linux-x.y.z.tar.gz`.

For the .tar files, use: `tar xpvf linux-x.y.z.tar`.

For the .tar.bz2 files, use: `bzip2 linux-x.y.z.tar.bz2`.

We explain the commands and their respective options in Step 5.



**Figure 10.4** Kernel numbering convention.

### Step 3. Check Space Requirements for the New Kernel and the New Source Tree

You will need an additional 55MB or so if you want to add the new kernel and source tree without touching the existing one. We recommend that you try to create a new “parallel” structure so that you do not interfere with the existing kernel structure for now. Once the new kernel is up and running satisfactorily, you can remove the existing structure. In extreme circumstances, you can just overwrite the existing structure with the new kernel. This strategy is not recommended, however.

To check available space, use the `df` command as follows:

```
[root@HostA /root]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda6        3.0G 1016M  1.8G   36% /
/dev/hda1         53M   3.5M   46M    7% /boot
```

### Step 4. Preserve the Old Source Tree

In this step, we discuss preserving the old source tree by renaming the directories and removing the links to the old Linux source tree.

First, check to see that you put the new tar file in the correct directory before starting. In the following example, we see that the new tar file is called *linux-2.4.5.tar.gz* in the */usr/src* directory:

```
[root@HostA /root]# ls -l /usr/src
total 25956
lrwxrwxrwx    1 root    root          11 Jul  3 13:26 linux-2.4 ->
linux-2.4.2
drwxr-xr-x   16 root    root        4096 Jul  3 13:26 linux-2.4.2
-rw-rw-r--    1 linda   linda    26534489 Jul  4 11:44 linux-2.4.5.tar.gz
drwxr-xr-x    7 root    root        4096 Jul  3 13:25 redhat
```

When you unpack this file, it will automatically create or overwrite any existing directory called *Linux* or any link thereto. Older versions of Red-Hat created a link between the *Linux* and *linux-2.4-2* (for example) directory. In such a case, you have to remove the link and rename the *linux-2.4-2* directory to something specific. In newer versions, the link is specific, such as *linux-2.4-5*.

Therefore in earlier versions, if we do not want the existing directory overwritten, we need to unlink it and remove the link to the directory called *linux*. Remove the link or rename the existing directory to keep it separate. It is customary to rename the *linux* directory by using the current

kernel revision number, which you can get by executing the `uname -r` command. In this example, you can see that this action is not necessary.

Now, we should have the `/usr/src` directory with no directory named *linux* in it. When the tar file is unpacked, it will create a new directory called *linux*.

### Step 5. Unpack Archived tar Files

Now, you can unpack the archived tar file named *linux-2.4.5.tar.gz*. We will use the `tar` command. The tar program is an archiving utility that stores files in a compressed format.

We will use `tar` with several of its options, as follows:

- x      Extract files from an archived tar file.
- p      Preserve all the permission/protection information appended to the files in the archive file.
- v      List individual files as they are unpacked (“verbose” option).
- f      Indicate which archive tar file to unpack.
- z      Filter archive through gzip as tar unpacks the .tar.gz-formatted tar file.

First, copy or move the file to the `/usr/src` directory. Then, because we are unpacking a .tar.gz file, we use all the options:

```
[root@HostA src]# # cd /usr/src
[root@HostA src]# tar xzpvf linux-2.4.5.tar.gz
```

Otherwise, if we had a .tar file, we would only use four of the options as follows:

```
# tar xpvf linux-2.4.5.tar
```

### Step 6. Create a Link to the New Linux Source Tree

This step is more of a recommendation than a mandatory task, but taking it will make the next few utilities behave differently. Do not create a link after this point, because it will be difficult. For example, the next few make commands will place and organize module and script information based on the link created here.



First, rename the directory named `linux` to `linux-new_revision_number` (in our case, it will be `linux-2.4.5`) by using the following command:

```
[root@HostA src]# ls
core  linux  linux-2.4  linux-2.4.2  linux-2.4.5.tar.gz  redhat
[root@HostA src]# mv linux linux-2.4.5
[root@HostA src]# ls
core  linux-2.4  linux-2.4.2  linux-2.4.5  linux-2.4.5.tar.gz  redhat
```

We want to create a link called `linux` to the `linux-2.4.5` directory so that the compile procedures will find the new directory and not the old one. Therefore, we enter

```
[root@HostA src]# ls -l
total 25960
drwxr-xr-x  14 1046 db2iadml  4096 May 25 19:12 linux
lrwxrwxrwx   1 root  root      11 Jul  3 13:26 linux-2.4 ->
linux-2.4.2
drwxr-xr-x  16 root  root      4096 Jul  3 13:26 linux-2.4.2
-rw-rw-r--   1 linda linda  26534489 Jul  4 11:44 linux-2.4.5.tar.gz
drwxr-xr-x   7 root  root      4096 Jul  3 13:25 redhat
[root@HostA src]# rm linux-2.4
rm: remove 'linux-2.4'? y
[root@HostA src]# mv linux linux-2.4.5
[root@HostA src]# ln -s linux-2.4.5 linux-2.4
[root@HostA src]# ls -l
total 25960
lrwxrwxrwx   1 root  root      11 Jul  4 11:58 linux-2.4 ->
linux-2.4.5
drwxr-xr-x  16 root  root      4096 Jul  3 13:26 linux-2.4.2
drwxr-xr-x  14 1046 db2iadml  4096 May 25 19:12 linux-2.4.5
-rw-rw-r--   1 linda linda  26534489 Jul  4 11:44 linux-2.4.5.tar.gz
drwxr-xr-x   7 root  root      4096 Jul  3 13:25 redhat
```

### **Step 7. Check for .o Files and Dependencies in the Source Tree**

We are now ready to remove any old `.o` files. The `make mrproper` command is a script that will find all old, incompatible object code (that is, `.o`) files that correspond to the version 2.4.2 code and systematically remove them from the `/usr/src/linux` tree. This script has to be run from the new `/usr/src/linux` directory. It will also remove information from a series of other related directories, as well. This script will make note of the name of the path and apply it to its logic. This script will also remove any old configuration files. So, if you are wanting to maintain your existing `.config` file

(which has all of your previous kernel settings in it), you should copy that file out of this directory now.

Let's invoke `make mrproper` now. Please note two things: You need to be in the `/usr/src/linux-2.4.5` directory to run `make mrproper`, and there will likely be a lot of output that will vary from what we show you here. Each system is configured differently, which results in different responses to this command.

```
# cd /usr/src/linux-2.4.2
# cp .config /usr/src/linux-2.4.5/.config-2.4.2
# cd /usr/src/linux-2.4.5
# make mrproper
make[1]: Entering directory '/usr/src/linux/arch/i686/boot'
rm -f tools/build
rm -f setup bootsect zImage compressed/vmlinux.out
rm -f bsetup bbootsect bzImage compressed/bvmlinux.out
make[2]: Entering directory '/usr/src/linux/arch/i686/boot/
compressed'
rm -f vmlinux bvmlinux_tmp_*
make[2]: Leaving directory '/usr/src/linux/arch/i686/boot/compressed'
make[1]: Leaving directory '/usr/src/linux/arch/i686/boot'
rm -f kernel/ksyms.lst include/linux/compile.h
find . -name '*.oas]' -type f -print | grep -v | xargs rm -f
rm -f core 'find . -type f -name 'core' -print'
rm -f core 'find . -name f -name '*.flags' -print'
rm -f vmlinux System.map
rm -f tmp*
rm -f drivers/char/consolemap_deftbl.c drivers/video/promcon_tbl.c
rm -f drivers/char/conmakehash
rm -f drivers/pci/devlist.h drivers/pci/classlist.h drivers/pci/
gen-devlist
rm -f drivers/sound/bin2hex drivers/sound/hex2hex
rm -f drivers/atm/fore200e_mkfirm drivers/atm/
{pca,sba}*{.bin,.bin1,.bin2}
rm -f net/khttpd/make_times_h
rm -f net/khttpd/times.h
rm -f submenu*
rm -rf modules
```

## Step 8. Check Software Requirements

Inside your new `/usr/src/linux-2.4.5/Documentation` directory is a file called *Changes*. You have to read this file and locate a stanza of information called “Current Minimal Requirements.” This information is a listing of software revisions that you must have before attempting to compile a new kernel.

All software entities and their respective revision levels are listed here, as are the various commands (for example, `gcc -version`, `ld -v`, `chsh -v`, and so on) that you can enter at the command line to check the respective version numbers. An excerpt of a sample file and stanza is shown next.

```
# cd /usr/src/linux/Documentation
# cat Changes | less
[root@HostA Documentation]# more Changes
Intro
=====
```

This document is designed to provide a list of the minimum levels of software necessary to run the 2.4 kernels, as well as provide brief instructions regarding any other "Gotchas" users may encounter when trying life on the Bleeding Edge. If upgrading from a pre-2.2.x kernel, please consult the Changes file included with 2.2.x kernels for additional information; most of that information will not be repeated here. Basically, this document assumes that your system is already functional and running at least 2.2.x kernels.

This document is originally based on my "Changes" file for 2.0.x kernels and therefore owes credit to the same people as that file (Jared Mauch, Axel Boldt, Alessandro Sigala, and countless other users all over the 'net).

The latest revision of this document, in various formats, can always be found at <<http://cyberbuzz.gatech.edu/kaboom/linux/Changes-2.4/>>.

Feel free to translate this document. If you do so, please send me a URL to your translation for inclusion in future revisions of this

--More--(9%)

.

.

.

Again, keep in mind that this list assumes you are already functionally running a Linux 2.2 kernel. Also, not all tools are necessary on all systems; obviously, if you don't have any PCMCIA (PC Card) hardware, for example, you probably needn't concern yourself with `pcmcia-cs`.

o Gnu C	2.91.66	# gcc --version
o Gnu make	3.77	# make --version
o binutils	2.9.1.0.25	# ld -v
o util-linux	2.10o	# fdformat --version
o modutils	2.4.2	# insmod -V
o e2fsprogs	1.19	# tune2fs
o reiserfsprogs	3.x.0j	# reiserfsck 2>&1 grep reiserf
sprogs		
o pcmcia-cs	3.1.21	# cardmgr -V

```

o PPP                2.4.0                # pppd --version
o isdn4k-utils       3.1pre1             # isdnctrl 2>&1|grep version

```

If you do not meet the current software requirements, you have some work to do. That is, if the results of entering the recommended commands provide software version numbers that are lower than those listed in the Changes file, you are expected to obtain the software and add it to your system. The software should be available in the form of RPM packages and can be added by using the same procedure as the one we used for the XFree86 RPM packages in Chapter 7.

We cannot emphasize enough the importance of this step. It takes time and patience, but it will ensure a successful installation. This process can be a very difficult and time-consuming part of creating your new kernel. If the instructions set out for you inside this new Changes file are not implemented, then you will be wasting valuable time from this point forward. Now, in some cases you might not require each and every one of these new pieces of code but rather only the ones that are applicable to your specific configuration (for example, your PC might not require PCMCIA-CS support). In our experience, this job is the task that when ignored or undertaken without due attention causes most upgrades to fail. If you are lucky, it just will not compile. If you are unlucky, it will compile something weird into your code, and it will drive you nuts from this time forward.

### Step 9. Configure the Kernel

This job is probably the most involved task, and you have to get it right. We do not provide steps here because we have already dedicated Chapter 9 to it. At this point, you should refer to Chapter 9.

You can return to configuring the kernel as many times as you need to until you feel you have it right. We recommend that you first run `make oldconfig` from the `linux-2.2.14-5` directory, in this case, which will show you the differences between the old and new kernels.

The `make oldconfig` command will attempt to configure the new kernel from the old configuration file. It will prompt you only for the new features. In earlier versions, to use `oldconfig` you have to preserve your previous configuration options. You have to copy your previous `/usr/src/linux-2.2.xx-x.0/.config` file to the new `/usr/src/linux` directory.

If you have an earlier version and you discover that you have no `.config` file, go to the directory containing the old source files (in this case, the `linux-2.2.xx-x` directory) and run `make config` at least once to generate a new `.config` file in that directory. Then, you can copy it over. We have used this technique in the exercise at the end of this chapter. You can also use

this procedure as a check to determine whether all new features added to the kernel are going to be available. As soon as you see this information, you should go immediately to make `xconfig` to continue configuration of the kernel.

We have already alluded to the fact that you also can use `make config` instead of `make xconfig`. You can even use `make menuconfig`. All these utilities do the same thing, but `make config` and `make menuconfig` require more effort and patience to see what is going on. The presentations are text and are overwhelming to the novice. We recommend that you use `make oldconfig` to preserve your existing configuration until you have this step down to a “mad science.”

```
[root@HostA linux-2.4.5]# make oldconfig
Ä
*** End of Linux kernel configuration.
*** Check the top-level Makefile for additional configuration.
*** Next, you must run 'make dep'.
```

### **Step 10. Prepare for Compilation: make dep and make clean**

The `make dep` command takes the `.config` file and builds a corresponding dependency tree. If this command does not return an error code of 0, return to `make xconfig` if that is the utility that you used originally. Otherwise, use the two others mentioned in the previous section and rectify any errors before proceeding to the next task.

The `make clean` command removes all files related to any previous compiles to ensure that you do not mangle or inherit previously compiled information.

Alas, many system managers tend to proceed past this point without reading their error messages. We recommend that the results of running these scripts are redirected to a file or that those error lines are `grep'd` and examined before proceeding. In the next example, the program complains about missing directories for some drivers that we do not wish to have anyway. These will not prevent us from compiling, but had we needed these drivers and missed these lines, we would have been in trouble—for a while, at least. In other words, you can compile a kernel containing errors as long as you know the exact nature of the errors and you are certain that they are irrelevant.

An example of the `make dep` command follows. Note that you must execute it from the `/usr/src/linux` directory or it will not work.

```
[root@HostA linux-2.4.5]# make dep;make clean > /tmp/depclean.out
```

You will notice that you get much more output than we have listed here.

```
make[2]: Leaving directory '/usr/src/linux-2.4.5/arch/i386/mm'
make -C arch/i386/lib fastdep
make[2]: Entering directory '/usr/src/linux-2.4.5/arch/i386/lib'
/usr/src/linux-2.4.5/scripts/mkdep -D__KERNEL__ -I/usr/src/linux-
2.4.5/include -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer -fno-
strict-aliasing -pipe -mpreferred-stack-boundary=2 -march=i686 --
checksum.S dec_and_lock.c delay.c getuser.S iodebug.c memcpy.c mmx.c
old-checksum.c putuser.S strstr.c usercopy.c > .depend
make[2]: Leaving directory '/usr/src/linux-2.4.5/arch/i386/lib'
make[1]: Leaving directory '/usr/src/linux-2.4.5'
make update-modverfile
make[1]: Entering directory '/usr/src/linux-2.4.5'
/usr/src/linux-2.4.5/include/linux/modversions.h was updated
make[1]: Leaving directory '/usr/src/linux-2.4.5'
bash: /tmp/depcclean.out: No such file or directory
```

### Step 11. Compile the Kernel: make bzImage

This step compiles the kernel and can take a while depending on the processing power of your system. We have seen this process take up to an hour. Be patient, and be prepared.

The commands for using zImage and bzImage follow. Note that you have to be in the `/usr/src/linux` directory to execute them or they will not work.

```
[root@HostA linux-2.4.5]# make zImage
gcc -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer -o scripts/split-
include scripts/split-include.c
scripts/split-include include/linux/autoconf.h include/config
gcc -D__KERNEL__ -I/usr/src/linux-2.4.5/include -Wall -Wstrict-
prototypes -O2 -fomit-frame-pointer -fno-strict-aliasing -pipe -
mpreferred-stack-boundary=2 -march=i686 -c -o init/main.o init/main.c
```

or

```
[root@HostA linux-2.4.5]# make bzImage
```

The difference between the two is simply the compression algorithm used to compress the file. No configuration issues exist, other than it might take a bit longer to uncompress a bzImage file than it will a zImage file during boot. The bzImage utility is an improvement over zImage only in terms of the compression ratio. This procedure will not overwrite the existing kernel. The new kernel will be placed in the `usr/src/linux-2.4.5/arch/i686/boot` directory and will be called bzImage, as shown:

```
[root@HostA boot]# ls -l | grep bz
-rw-r--r--  1 root    root      967412 Jul  2 09:39 bzImage
```

The new kernel file, *bzImage*, is added to the already-existing kernel files in the directory.

A limitation of Intel machines requires the kernel to be decompressed at startup time within the first 600–700KB of memory. Otherwise, you might get an error indicating that the kernel is too large. If you get a message like that, use *bzImage* instead of *zImage* to compile a smaller kernel so that you can work around this limitation.

### Step 12. make modules and make modules\_install

Assume that during the `make xconfig` procedure (see the beginning of this chapter) you chose to use the modules you need for `make modules` and `make modules_install`. Executing these commands in this step will overwrite your existing module information if you do not rename the directory. The modules reside in the `/lib/modules` directory as `/usr/modules/<kernel_release>`. We recommend that if you have an earlier release of Linux that you rename the directory prior to issuing these commands. You must execute the `make modules` and `make modules_install` commands from the `/usr/src/linux` directory or they will not work.

```
[root@HostA 2.4.5]# cd /usr/src/linux-2.4.5
[root@HostA 2.4.5]# make modules
[root@HostA 2.4.5]# make modules_install
[root@HostA src]# ls -l /lib/modules
total 2
drwxr-xr-x  4 root    root      1024 Jun 18 07:09 2.4.2-2
drwxr-xr-x  4 root    root      1024 Jul  2 09:49 2.4.5
```

**TIP** If you want to make a boot disk (without the root file system or LILO), insert a floppy diskette in your A: drive and execute the `make disk` command. Now, you have a diskette with the kernel on it to test without changing your system. When the kernel is what you want, then proceed to put this new kernel in place on your machine for future use.

The `make modules` command creates the modules. The `make modules_install` command installs the modules in the directory. The new modules should be in a directory with the version number of the kernel you just compiled.

If errors occur, you have probably chosen some conflicting modules. To resolve these issues, go back to the `oldconfig` or `xconfig` stage and rework

the kernel options again and again as many times as you need. At some point in this process, you might want to resort to paper and pen to map out an objective that does not contain conflicts. You should use the Help option within the kernel configuration tool to carefully research each item that you are adding to or removing from the new kernel. We are also assuming that you still have access to the original *.config* file, which corresponds to the previous kernel version. The scientific method works well here. Consider changing only one thing at a time to find out where the break is.

### Step 13. Copy bzImage Kernel and System.map Files to Correct Directories

The new bzImage kernel file in */usr/src/linux/arch/i686/boot* should be copied to the */boot* directory. We recommend that you rename this file so that it does not overwrite the old kernel file, and then relink it.

```
[root@HostA boot]# cd /usr/src/linux-2.4.5/arch/i386/boot
[root@HostA boot]# ls bz*
bzImage
[root@HostA boot]# cp bzImage /boot/vmlinuz-2.4.5
[root@HostA boot]# ls -l /boot/vm*
-rwxr-xr-x 1 root root 2280836 Apr  8 18:57 /boot/vmlinuz-2.4.2-2
lrwxrwxrwx 1 root root      15 Jun 18 07:09 /boot/vmlinuz-> vmlinuz
-2.4.2-2
-rw-r--r-- 1 root root 781806 Apr  8 18:57 /boot/vmlinuz-2.4.2-2
-rw-r--r-- 1 root root 967412 Jul  2 10:17 /boot/vmlinuz-2.4.5
```

Note that the second command in the preceding listing copies and renames simultaneously so that the older version is not clobbered. In addition, the *ln* command gives it the generic name *vmlinuz* that LILO will use to refer to it.

```
[root@HostA /boot]# rm vmlinuz
rm: remove 'vmlinuz'? y
[root@HostA /boot]# ln -s vmlinuz-2.4.5 vmlinuz
[root@HostA /boot]# ls -l vm*
-rwxr-xr-x 1 root root 2280836 Apr  8 18:57 vmlinuz-2.4.2-2
lrwxrwxrwx 1 root root      13 Jul  2 10:25 vmlinuz ->
vmlinuz-2.4.5
-rw-r--r-- 1 root root 781806 Apr  8 18:57 vmlinuz-2.4.2-2
-rw-r--r-- 1 root root 967412 Jul  2 10:17 vmlinuz-2.4.5
```

You will also have to replace *System.map*, although some systems have a static *System.map* file (in which case, you are OK here). We have assumed



that you have a symbolic link named *System.map* that has to be modified and copied.

When the kernel was compiled, a new *System.map* file was created that must also be put in place. Otherwise, when you reboot, the kernel will give you a “missing *System.map*” error.

The new *System.map* file will be in the top-level directory called */usr/src/linux*. This file should have the same creation time and date as the *bzImage* file. Again, we recommend that you rename the *System.map* file and create a link to it so as not to overwrite the old *System.map* file.

```
[root@HostA linux-2.4.5]# pwd
/usr/src/linux-2.4.5
[root@HostA linux-2.4.5]# ls -l Sy*
-rw-r--r--  1 root    root      456588 Jul  2 09:39 System.map
[root@HostA linux-2.4.5]# cp System.map /boot/System.map-2.4.5
[root@HostA linux-2.4.5]# cd /boot
[root@HostA /boot]# ls -l Sys*
lrwxrwxrwx  1 root    root          18 Jul  2 09:32 System.map ->
System.map-2.4.2-2
-rw-r--r--  1 root    root      410990 Apr  8 18:57 System.map-2.4.2-2
-rw-r--r--  1 root    root      456588 Jul  2 10:32 System.map-2.4.5
[root@HostA /boot]# rm System.map
rm: remove 'System.map'? y
[root@HostA /boot]# ln -s System.map-2.4.5 System.map
[root@HostA /boot]# ls -l Sy*
lrwxrwxrwx  1 root    root          16 Jul  2 10:34 System.map ->
System.map-2.4.5
-rw-r--r--  1 root    root      410990 Apr  8 18:57 System.map-2.4.2-2
-rw-r--r--  1 root    root      456588 Jul  2 10:32 System.map-2.4.5
[root@HostA /boot]# ls -l module*
lrwxrwxrwx  1 root    root          19 Jun 18 07:09 module-info ->
module-info-2.4.2-2
-rw-r--r--  1 root    root      13598 Apr  8 18:57 module-info-2.4.2-2
[root@HostA /boot]# cp module-info-2.4.2-2 module-info-2.4.5
[root@HostA /boot]# rm module-info
rm: remove 'module-info'? y
[root@HostA /boot]# ln -s module-info-2.4.5 module-info
[root@HostA /boot]# ls -l module*
lrwxrwxrwx  1 root    root          17 Jul  2 10:50 module-info ->
module-info-2.4.5
-rw-r--r--  1 root    root      13598 Apr  8 18:57 module-info-2.4.2-2
-rw-r--r--  1 root    root      13598 Jul  2 10:49 module-info-2.4.5
```

#### Step 14. Optional: Create a RAM Disk File

The purpose of using an initial RAM disk file is to enable a modular kernel to have access to modules that it might need to boot from before the kernel

has access to the device where the modules normally reside. Not all systems require you to make an `initrd` or RAM disk file, but for older kernel versions and older hardware, this action can be paramount. For example, we have an old SCSI system that cannot find the drive unless we preload the Adaptec 1520 controller. We hope that you have better equipment, but a system of ours refuses to die. We have appropriately named it Christine. To make the RAM disk, which you will call `initrd-2.4.5.img`, you use the `mkinitrd` command with the new kernel as its argument, as seen in the following example:

```
# cd /boot
# mkinitrd /boot/initrd-2.4.5.img 2.4.5
# ls -l init*
-rw-r--r-- 1 root root 278788 Jul 1 15:08 initrd-2.4.5.img
```

When finished, you will have a RAM disk file in the `/boot` directory called `initrd-2.5.img`, which you will point to from the `/etc/lilo.conf` file.

### Step 15. Modify the `/etc/lilo.conf` File to Point to New

Kernel without and with a RAM Disk File Change directory to `/etc`. We reproduce the contents of the `lilo.conf` file here:

```
[root@HostA /boot]# cat /etc/lilo.conf
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=linux

image=/boot/vmlinuz-2.4.5
    label=linux
    read-only
    root=/dev/hda6
```

Using your favorite text editor, make the following changes to your `lilo.conf` file if you are using a RAM disk:

```
[root@HostA /boot]# cat /etc/lilo.conf
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
```

```
prompt
timeout=50
message=/boot/message
linear
default=linux

image=/boot/vmlinuz-2.4.5
    initrd=/boot/initrd-2.4.5.img
    label=linux
    read-only
    root=/dev/hda6
```

Before you exit this section, make sure that the files you are pointing to in the `image=` statements are in the boot directory. The `lilo -v` command should tell you, but only you know if you copied the correct one.

### **Step 16. Update the MBR**

Now, run `lilo -v` to update the MBR with the new LILO information. Remember that if you make any changes to the `/etc/lilo.conf` file, you have to rerun `lilo -v` to update the master boot record.

```
# lilo -v
LILO version 21, Copyright 1992-1998 Werner Almesberger Reading boot
sector from /dev/had Merging with /boot/boot.b
Boot image: /boot/vmlinuz-2.4.5
Added linux *
/boot/boot.0300 exists - no backup copy made.
Writing boot sector.
```

### **Step 17. Shut Down and Test System: Rescue If Necessary**

Shut down and reboot the system with the following command:

```
# shutdown -r now
```

To boot a system that has a Linux version equivalent to those older than Red Hat Linux 6.1 or 6.2, you will likely need two diskettes: a boot diskette and a rescue diskette. If you have Red Hat Linux 6.1 or 6.2 and the system will boot from a CD-ROM, then you only need the Linux CD. If booting from the CD-ROM drive is not an option, then you will need the boot diskette.

Because Step b has to be taken quickly, read this procedure before you begin.

1. Boot Linux from the boot diskette (that is, the one with the boot image), unless your system is capable of booting from the CD.
2. Quickly, when the `boot :` prompt appears, type `rescue`—which will load the kernel from the diskette or the CD-ROM, whichever is applicable in this situation.
3. Follow the prompts. You might need to replace a boot disk with a rescue disk when prompted to do so, depending on your configuration.
4. When you get a shell prompt, you can try to mount your drives.
5. Create an empty mount point.

```
# mkdir /mnt/linux
# mount -t ext2 /dev/hda1 /mnt/linux
```

Good luck.

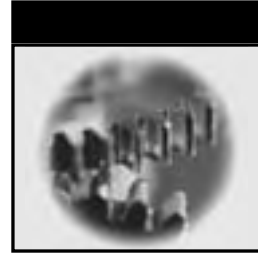
If you are desperate to boot, you can also try to boot from the Linux installation CD-ROM.

If you are super desperate to boot, you can try to do so from DOS via accessing the CD-ROM and using the `autoboot` option under the `dosutils`.

### **Step 18. Check New Kernel**

There are three ways to check on your new kernel. First, make sure the new login herald reflects the new kernel revision. Then, when you are logged in, issue a `uname -a` command and make sure that it, too, reflects the new revision. At this point, you should proceed to check all applications to ensure that they are responding correctly and as expected. If all three of these checks work, congratulations are in order.





# Index

- accelerated graphics port (AGP), in video configuration, 126
- accelerators, in video configuration, 128
- access control, file sharing, NFS, 114, 117–118
- active partition, 15
- adapters, video, 126–128, 157, 162
- advanced power management BIOS support
  - option, in kernel configuration, 237
- authentication
  - IDE system installation, 58
  - SCSI system installation, 88, 91
- automatic mount, 116
- automatic partitioning, 22–23
- bandwidth, in video configuration, 141–143
- bash shell, 252
- basic input/output system (BIOS), 1–2
  - in boot process, 15
  - in video configuration, 156
  - LinuxLoader (LILO) and, 17
- binary files, binaries, NFS installation, 103
- block devices option, in kernel configuration, 237
- boot directory, 27–29
- boot disks, 4, 64, 94–95
  - creating, 109–110
  - DOS/Windows, 36–38
  - emergency boot disk for, 220, 225–226, 257–258
  - IDE system installation, creating, 36–40
  - in source tree kernel install and, 257–258, 268
  - NFS installation, 109–110
  - RAM disk, 222, 226
  - UNIX, 36, 38–40
- boot files, 22, 25, 49–50, 75
- boot images, problems, 40
- Boot Manager, 14, 16
- boot message
  - IDE system installation, 44–46
  - SCSI system installation, 70
- boot process, 14–16
  - active partition in, 15
  - basic input/output system (BIOS) in, 15
  - boot directory in, 28–29
  - boot disk creation for (*See* boot disks)
  - boot files in, 22, 25, 49–50, 75
  - boot images and, problems in, 40
  - Boot Manager in, 14, 16
  - boot sector in, 15

- branching in, 15
- CMOS in, 15
- getty process in, 15
- inittab file in, 15
- installation disk boot problems in, 42–43
- LinuxLoader (LILO) in, 16–21
- Master Boot Record (MBR) in, 18, 20, 50–51, 84, 222, 226
- Partition Magic in, 16
- Plug-and-Play parameters in, 15
- power on self test (POST) in, 15
- RAM disk support file creation for, 222, 226
- random access memory (RAM) in, 15
- read only memory (ROM) in, 15
- root file system and, 15
- boot sector, 15
- branching, in boot process, 15
- buses and slots, in video configuration, 126, 128–132
- bzimage, in source tree kernel install, 269–270
- cathode ray tube (CRT) displays, 126
- CD-ROM
  - installation from (*See* CD-ROM installations), 35
  - partitioning, 32
- CD-ROM installations, 35–101
  - authentication configuration in, 58, 88, 91
  - boot disk creation for, 36–40, 64, 94–95
  - boot images and, problems in, 40
  - boot message for, 44–46, 70
  - boot problems in, 42–43
  - buttons in, 41
  - check boxes in, 41
  - confirming and testing, 100–101
  - custom, 49, 70
  - development packages in, 91
  - disk space available in, 61
  - documentation of, 101
  - DOS/Windows connectivity in, 91
  - driver problems in, 43–44
  - entering information in, 41–42
  - expert mode for, 45
  - firewall configuration in, 53–54, 86
  - FTP and, 59
  - GNOME in, 92
  - GNU Public License information and, 36
  - IDE system installation, 42–68
    - authentication configuration in, 58
    - boot message for, 44–46
    - boot problems in, 42–43
    - custom, 49
    - DOS/Windows connectivity in, 59
    - expert mode for, 45
    - firewall configuration in, 53–54
    - FTP and, 59
    - games in, 59
    - GNOME in, 59
    - GNOME versus KDE installation
      - selection in, 48–49
    - graphical vs. text mode message in, 45
    - IP addresses in, 51–53
    - kernel development in, 59
    - keyboard specification in, 46–48
    - language selection in, 46
    - language support in, 55
    - LinuxLoader (LILO) installation/
      - configuration in, 50–51
    - Master Boot Record (MBR) install in, 50–51
    - master slave configurations, 43
    - monitor setup in, 64–65
    - mouse selection in, 54–55
    - network configuration in, 51–53
    - networked workstations in, 59
    - NFS Server in, 59
    - non-root user addition in, 57–58
    - package group selection in, 58–61
    - partitioning in, 49–50
    - printer support in, 59
    - process screen and installation of selected
      - items in, 62–64
    - rescue mode for, 45–46
    - root password selection in, 56–57
    - stepping through installation of, 44–68
    - Telnet and, 59
    - Telnet setting in, 57
    - time zone configuration in, 55–56
    - troubleshooting, 43–44
    - utilities in, 59
    - video card problems in, 44
    - video configuration in, 61–62, 65–67
    - Welcome screen in, 48
    - X Window System and, 48, 59, 67–68
  - initiating, 36–42
  - invoking installation program in, 40–41
  - IP addresses in, 51–53, 85–86
  - KDE in, 91
  - kernel development package in, 91
  - keyboard specification in, 46–48, 70
  - language selection in, 46, 70
  - language support in, 55, 88
  - linuxconf in, 92
  - LinuxLoader (LILO) installation/
    - configuration in, 50–51, 83–85
  - logs for, reviewing, 101
  - Master Boot Record (MBR) install in, 50–51
  - master slave configurations, 43

- monitor setup in, 64–65, 95–96
- mouse selection in, 54–55, 86–88
- multimedia support in, 91
- network card problems in, 43–44
- network configuration in, 51–53, 85–86
- networked workstations in, 91
- networking and, 100
- NFS Server in, 91
- non-root user addition in, 57–58, 88
- package group selection in, 58–61, 91–92
- partitioning in, 49–50, 71–83
- PCMCIA installation and, 36
- printer in, 91
- process screen and installation of selected items in, 62–64, 93–94
- RAID array in, 68, 74–83
- reconfiguring the system in, 101
- rescue mode for, 45–46
- root password selection in, 56–57, 88
- screens or windows (dialog boxes) in, 40–41
- scroll and select dialog boxes in, 41
- SCSI system installation, 68–101
  - authentication configuration in, 88, 91
  - boot message for, 70
  - custom, 70
  - development packages in, 91
  - DOS/Windows connectivity in, 91
  - firewall configuration in, 86
  - GNOME in, 92
  - IP addresses in, 85–86
  - KDE in, 91
  - kernel development package in, 91
  - keyboard specification in, 70
  - language selection in, 70
  - language support in, 88
  - linuxconf in, 92
  - LinuxLoader (LILO) installation/
    - configuration in, 83–85
  - Master Boot Record (MBR) install in, 50–51
  - mouse selection in, 86–88
  - multimedia support in, 91
  - network configuration in, 85–86
  - networked workstations in, 91
  - NFS Server in, 91
  - non-root user addition in, 88
  - package group selection in, 91–92
  - partitioning in, 71–83
  - printer in, 91
  - process screen and installation of selected items in, 93–94
  - RAID array in, 68, 74–83
  - root password selection in, 88
  - time zone configuration in, 88
  - troubleshooting, 99–100
  - user interface/X in, 92
  - utilities in, 92
  - video configuration in, 92–93, 96–98
  - Web server in, 91
  - Welcome screen in, 70
  - X Window System in, 91, 98–99
  - XFree86-xf86cfg in, 92
- Session Message Block (SMB) installation
  - and, 36
- TCP/IP configuration in, 100
- Telnet and, 59
- test input dialog boxes in, 41
- text menu vs. GUI, 35–36
- time zone configuration in, 55–56, 88
- troubleshooting, 43–44, 99–100
- user interface/X in, 92
- utilities in, 92
- video card problems in, 44
- video configuration in, 61–62, 65–67, 92–93, 96–98, 100–101
- Web server in, 91
- Welcome screen in, 48, 70
- X Window System and, 48, 67–68, 91, 98–99
- XFree86-xf86cfg in, 92
- central processing unit (CPU) usage, 8, 126
- chipsets, in video configuration, 128, 130, 156
- client/server, file sharing, NFS, 104, 109–111, 117, 121–123
- clock, dot clock or VGA memory clock, 132, 141–143
- CMOS
  - in boot process, 15
  - time zone configuration in, 55–56, 88
- code maturity level options, in kernel
  - configuration, 231–232
- color depth, in video configuration, 160
- configuration, LinuxLoader (LILO), etc/lilo/
  - conf file, 18–21, 50, 83–85
- controllers, 3
- database servers, memory requirements, 10
- default Linux partition, 12
- deflection yoke, in displays and monitors, 134–136
- depmod, in source tree kernel install, 252–253
- desktop, in video configuration, 168–171
- dev/hda logical partitions, 23–26, 31
- development packages, 91
- digital to analog converters (DACs), in video
  - configuration, 132
- directories
  - file sharing, NFS, 113, 120, 122–123
  - in source tree kernel install, 269–270



- logical partitions (dev/hda), 23–26, 31
- mounting, for file sharing, 113–117, 122–123
- NFS installation, 108–109
- partitioning, 27
- printer configuration and, spool, 185–186
- proc/ide directory in, 31
- proc/scsi/scsi in, 32
- root file system and, 15, 21–22
- RPM packages, 200
- separate partitions for, pros and cons, 27–30
- XF86Setup, 211
- Disk Druid, 12–13, 27
- disk operating system (DOS), 4
- disk paging, 7–11
- displays and monitors, 3, 157
  - custom monitor setup in, 158–159
  - deflection yoke in, 134–136
  - electron guns in, 132–136
  - frames in, 134–136
  - horizontal scan frequency (HSF) in, 135, 159–160
  - horizontal synchronization pulse (HSP) in, 135–139, 159–160
  - IDE system installation and setup in, 64–65
  - phosphor screen in, 133, 145
  - refresh rate in, 136
  - SCSI system installation and setup in, 95–96
  - shadow mask in, 133, 145
  - synchronization in, 134–136, 144, 159, 160
  - vertical scan frequency (VSF) in, 136
  - vertical synchronization pulse (VSP) in, 135–139, 144, 159, 160
- documentation, NFS installation, 101
- DOS/Windows
  - boot disk creation for, 36–38
  - connectivity, IDE system installation, 59, 91
  - dot clock or VGA memory clock, 132, 141–143
  - dot pitch, 144–147
- drivers, 2
  - IDE system installation problems in, 43–44
  - integrated drive electronics (IDE), 2
  - kernel upgrades using RPMs, 214
  - Plug-and-Play support in, 2
  - printer configuration and, 179–180, 183, 189–190
- dummy net driver support option, in kernel configuration, 243
- e-mail, memory requirements, 10
- electron guns, in displays and monitors, 132–136
- emergency boot disk, 220, 225–226, 257–258
- etc/hosts.lpd file, remote printing, 190–192
- etc/lilo.conf file, 18–21, 50, 83–85
  - in source tree kernel install and, 271–272
- Ethernet (10 or 100Mbit) option, in kernel configuration, 243–244
- expert mode, IDE system installation, 45
- extended partition, 12, 25–26
- fdisk
  - partitioning, 12–13
  - swap partition viewing using, 8–9
- file sharing, NFS, 112–119
  - access control in, 114, 117–118
  - automatic mount in, 116
  - casual NFS mount for, 103
  - client setup for, 121–123
  - client/server communications in, 117
  - daemons in, start/stop, 114–115
  - directory export to, 120
  - directory organization in, 113, 122–123
  - mount point for, 115–116, 121
  - mounting directory for, 113–117, 122–123
  - permissions and, 117–119
  - server setup for, 120–121
  - temporary mount in, 116
  - timing parameters for, 117
  - uids and gids in, 112–113, 118–119
  - Windows to/from Linux system example of, 119–123
  - Xlink Technology for, 119–123
- filesystems, in kernel configuration, 247
- firewall
  - IDE system installation, 53–54
  - SCSI system installation, 86
- First Nondestructive Interactive Partition Splitting (FNIPS) program, 12
- formatting, partition, 80–83
- frames, in displays and monitors, 134–136
- free space, 13
- freeware status of Linux, 36
- FTP
  - IDE system installation, 59
  - RPM packages and, 200–202
- games, IDE system installation, 59
- getty process, 15
- GNOME, 4, 92
  - IDE system installation versus KDE, 48–49
  - IDE system installation, 59
  - in video configuration and, 168
- GNU Public License information, 36
- group ids (gids), file sharing, NFS, 112–113, 118–119
- hard disk partitioning (*See* partition planning and preparation), 3, 7–34
  - multiple hard drive, 30
- hardware conflicts, 2

- hardware support, kernel upgrades using RPMs, 214
- high definition TV (HDTV), 149
- home directory, 27, 29
- horizontal scan frequency (HSF), 135, 144
- horizontal synchronization pulse (HSP), 135–139, 159–160
- Hwinfo utility, 4
- IDE system configuration, 2
  - authentication configuration in, 58
  - boot message for, 44–46
  - boot problems in, 42–43
  - custom, 49
  - DOS/Windows connectivity in, 59
  - expert mode for, 45
  - firewall configuration in, 53–54
  - FTP and, 59
  - games in, 59
  - GNOME in, 59
  - GNOME vs. KDE installation selection in, 48–49
  - graphical vs. text mode message, 45
  - IDE system installation, 42–68
  - installation troubleshooting, 43–44
  - iomem file in, 44
  - ioports file in, 44
  - IP addresses in, 51–53
  - irq file in, 44
  - kernel development in, 59
  - keyboard specification in, 46–48
  - language selection in, 46
  - language support in, 55
  - LinuxLoader (LILO) installation/
    - configuration in, 50–51
  - Master Boot Record (MBR) install in, 50–51
  - master slave configuration in, 31–32, 43
  - monitor setup in, 64–65
  - mouse selection in, 54–55
  - net file in, 44
  - network configuration in, 51–53
  - networked workstations in, 59
  - NFS Server in, 59
  - non-root user addition in, 57–58
  - package group selection in, 58–61
  - partitioning in, 49–50
  - partitioning, geometry of, 30–32
  - printer support in, 59
  - proc/ide directory in, 31
  - rescue mode for, 45–46
  - root password selection in, 56–57
  - screen and installation of selected items in, 62–64
  - stepping through installation of, 44–68
  - Telnet setting in, 57, 59
  - time zone configuration in, 55–56
  - utilities in, 59
  - video configuration in, 61–62, 65–67
  - Welcome screen in, 48
  - X Window System and, 48, 59, 67–68
- image data transfer rate, 130, 131
- include IDE/ATA-2 disk support option, in
  - kernel configuration, 239
- include IDE/ATAPI options, in kernel
  - configuration, 239
- Industry Standard Architecture (ISA), 42, 126
- init process, 15
- inittab file, 15
- insmod module, in source tree kernel
  - install, 253
- installation
  - memory requirements for, 4, 9–11
  - preinstallation inventory of system for, 1–6
- Integrated Drive Electronics (*See* IDE
  - system configuration)
- interlacing, in video configuration, 147–149
- iomem file, 44
- ioports file, 44
- IP addresses, 6
  - IDE system installation, 51–53
  - NFS installation, 107, 110–111
  - SCSI system installation, 85–86
- IRQ conflicts, 4, 44
- irq file, 44
- K Desktop Environment, 4
- KDE systems, 91
  - IDE system installation vs. GNOME, 48–49
  - in video configuration and, 168
- Kerberos, IDE system installation, 58
- kernel (*See also* kernel configuration; kernel
  - install using source tree; kernel
    - upgrades using RPMs)
  - configuration of (*See* kernel configuration), 229–247
  - in boot process, 15
  - life cycle of, 251
  - monolithic vs. modular, 233, 250–255
  - RPM packages to upgrade (*See* kernel
    - upgrades using RPMs)
  - source tree build installation (*See* kernel
    - install using source tree)
- kernel configuration
  - advanced power management BIOS
    - support option in, 237
  - block devices option in, 237
  - code maturity level options in, 231–232
  - dimmed menu options in, 230–231
  - dummy net driver support option in, 243
  - Ethernet (10 or 100Mbit) option in, 243–244

- filesystems in, 247
- general setup options in, 235–237
- include IDE/ATA-2 disk support option in, 239
- include IDE/ATAPI options in, 239
- in source tree kernel install, 265–266
- kernel core option in, 236
- loadable module support in, 233–235
- loopback device support option in, 240
- menu and menu contents in, 230–247
- monolithic vs. modular kernels in, 233
- multiple device driver support option in, 240
- network block device support option in, 240
- network device support option in, 242–244
- networking options in, 240–242
- networking support option in, 236
- normal PC floppy disk support option in, 237–239
- Other IDE chipset support option in, 239
- parallel port support option in, 237
- PCI access mode option in, 236
- PCI quirks option in, 236
- PCI support option in, 236
- Plug-and-Play support in, 237
- processor types and features in, 232
- RAM disk support option in, 240
- save and exit in, 247
- SCSI emulation support option in, 239
- SCSI low level drivers in, 245–247
- SCSI support in, 244–247
- System V IPC option in, 236
- TCP/IP networking option in, 241
- UNIX domain sockets option in, 241
- xconfig files for, 229–230
- kernel core option, in kernel configuration, 236
- kernel development, IDE system installation, 59
- kernel development package, 91
- kernel install using source tree, 249–273
  - application check for, 273
  - bash shell and, 252
  - boot disk creation for, 257–258, 268
  - bzimage copy in, 269–270
  - compilation in, make bzimage command, 267–268
  - compilation in, using make dep and make clean commands, 266–267
  - copying old configuration for, using makeoldconfig method, 250
  - current kernel information and, 257
  - depmod for, 252–253
  - directories for, 269–270
  - disk space requirements for, 260
  - emergency boot disk for, 257–258
  - etc/lilo.conf file modification for, 271–272
  - failure to load module in, 255
  - finding a new kernel for, 256
  - insmod module in, 253
  - installing new kernel for, 256–273
  - kernel configuration in, 265–266
  - kerneld in, 251, 253
  - kmod module in, 253
  - life cycle of kernel and, 251
  - link to new source tree for, 261–262
  - LinuxLoader (LILO) and, 249
  - lockd module in, 253
  - login check for, 273
  - lsmod module in, 253
  - main modutils file for, lib/modules/xxx, 253–254
  - make modules command in, 268–269
  - make modules\_install command in, 268–269
  - master boot record (MBR) update for, 272
  - modprobe module in, 253
  - module utilities for, 252–253
  - modules in, 250–255
  - monolithic vs. modular kernels in, 250–255
  - nfsd module in, 253
  - o files and dependencies in source tree and, 262–263
  - old source tree preservation before, 260–261
  - RAM disk file for, 270–271
  - reinstall loadable modules in, 255
  - rmmod module in, 251, 253
  - software requirements for, 263–265
  - source code for, in tar file format, 258–259
  - sunrpc module in, 253
  - system.map files in, 269–270
  - testing, 272–273
  - unpacking archived tar files for, 261
  - upgrade issues with modules in, 254–255
  - versional differences in, 249
  - zlilo utility for, 249
- kernel upgrades using RPMs, 213–227
  - already installed package identification in, 216–218, 224–225
  - backup before, 216
  - check and review of, 223–224
  - comparing existing to downloaded packages for, 219, 225
  - current version recording in, 219–220, 225
  - disk space requirements for, 216, 224
  - driver updates and, 214
  - emergency boot disk for, 220, 225–226
  - FTP files for upgrading, 225
  - install and upgrade procedure for, 221–222, 226

- integrity check for, 218–219
- LinuxLoader (LILO) setup after, 223, 226–227
- new hardware support and, 214
- organizing packages for, 218–219
- partitioning for, 216
- RAM disk support file creation for, 222, 226
- reasons for, 214–215
- reboot and test, 224, 227
- Red Hat v7.x procedure, 215–224
- RedHat v6.x procedures, 224–227
- security issues and, 214–215
- kernel, 251, 253
- keyboard
  - IDE system installation, specification of, 46–48
  - SCSI system installation, specification of, 70
- kmod module, in source tree kernel install, 253
- language selection
  - IDE system installation, 46
  - SCSI system installation, 70
- language support
  - IDE system installation, 55
  - SCSI system installation, 88
- lib/modules/xxx, in source tree kernel install, 253–254
- life cycle, kernel, 251
- Lightweight Directory Access Protocol (LDAP), 58
- line printer (lpr) command in, 173–177, 181–185
- line printer spooler daemon (lpd) command, 173–177, 180–186, 190
- linuxconf, 92
- LinuxLoader (LILO), 14, 16–21
  - BIOS and, 17
  - etc/lilo/conf file for, 18–21, 50, 83–85
  - IDE system installation, 50–51
  - in source tree kernel install, 249, 271–272
  - Master Boot Record (MBR) in, 18, 20
  - new or upgraded kernel and, setup for, 223, 226–227
  - SCSI system installation, 83–85
- loadable module support, 233–235
- lockd module, in source tree kernel install, 253
- logical partitions (dev/hda), 12–13, 23–26, 31
- logs, NFS installation, 101
- loopback device support option, in kernel configuration, 240
- LPRng package, 174–176
- lsmod module, in source tree kernel install, 253
- main modutils file, in source tree kernel install, lib/modules/xxx, 253–254
- make bzimage command, 267–268
- make clean command, 266–267
- make dep command, 266–267
- make modules command, 268–269
- make modules\_install command, 268–269
- makeoldconfig method, 250
- Master Boot Record (MBR), 14, 18, 20
  - IDE system installation, 50–51
  - in source tree kernel install, updating, 272
  - kernel upgrades using RPMs, 222, 226
  - LinuxLoader (LILO) and, 18, 20
  - SCSI system installation, 50–51
- master slave configuration
  - IDE system installation, 43
  - partitioning, 31–32
- memory, 3
  - in source tree kernel install requirements, 260
  - kernel upgrades using RPMs requirements, 216, 224
  - paging in, 8–9
  - random access memory (RAM) in, 8–11
  - read only memory (ROM) in, 15
  - requirements, 4, 9–11, 61, 216, 224, 260
  - size of swap space in, 9–11
  - thrashing, 8
  - video adapter (VRAM), 126–128, 162
- mode, video mode, 127–129, 163–164
- modeline numbers and codes, in video configuration, 140
- modems, 3
- modprobe module, in source tree kernel install, 253
- modular kernel, 233, 250–255
- monitors (*See* displays and monitors; video)
- monolithic kernel, 233, 250–255
- mount point, 8–9
  - in file sharing, NFS, 115–116, 121
  - partitioning, 24
- mounting a directory
  - automatic, 116
  - for file sharing, 113–117, 122–123
  - temporary, 116
- mouse
  - IDE system installation, 54–55
  - SCSI system installation, 86–88
- multimedia support, 91
- multiple device driver support option, in kernel configuration, 240
- naming RPM packages, 197
- network block device support option, in kernel configuration, 240

- network card, 3
  - IDE system installation problems, 43–44
- network device support option, in kernel configuration, 242–244
- Network File System (*See* NFS)
- Network Information Services (NIS)
  - IDE system installation, 58
- Network Time Protocol (NTP), 56
- networked workstations, 59, 91
- networking, 4, 6
  - IDE system installation, 51–53
  - IP addresses, 6, 51–53, 85–86
  - NFS installation, 100, 104–105, 110–111
  - SCSI system installation, 85–86
- networking options, in kernel configuration, 240–242
- networking support option, in kernel configuration, 236
- NFS installation, 103–123
  - binary files, binaries in, 103
  - boot disk creation in, 109–110
  - casual mount for file sharing, 103
  - client setup in, 109–111
  - client/server and, 104
  - configuration details in, 105–111
  - copying files to hard disk in, 107–108
  - directory export to, 108–109
  - file sharing using, 112–119
  - image options in, 110
  - IP addresses in, 107, 110–111
  - network configuration in, 104–105
  - networking information for, 110–111
  - package for, 106
  - RPMs in, 103
  - security settings in, 108–109
  - server daemons in, 106–107
  - server information for client setup in, 111
  - server setup in, 105–109
  - standard build in, 103
  - target system boot in, 110
- NFS Server, 59, 91
- nfsd module, in source tree kernel install, 253
- normal PC floppy disk support option, in kernel configuration, 237–239
- NTFS filesystem, 247
- operating systems, 4
  - partitioning, multiple, 13
- Other IDE chipset support option, in kernel configuration, 239
- package group selection, 58–61, 91–92
- paging of memory, 8–9
- parallel port support option, in kernel configuration, 237
- Partition Magic, 12, 16
- partition managers, 13
- partitioning, 7–34
  - active partition in, 15
  - automatic partitioning in, 22–23
  - boot directory, 28–29
  - boot files and, 22, 25, 49–50, 75
  - Boot Manager code for, 14, 16
  - boot process and, 14–16
  - boot sector and, 15
  - CD-ROM and, 32
  - central processing unit (CPU) usage and, 8
  - default Linux partition in, 12
  - default partitioning behavior in, 22
  - directories created in, 27
  - Disk Druid for, 12–13, 27
  - disk paging in, 7–11
  - extended partition in, 12, 25–26
  - fdisk for, 12–13
  - First Nondestructive Interactive Partition Splitting (FIPS) program for, 12
  - formatting, 80–83
  - free space in, 13
  - home directory, 29
  - IDE drive geometry and, 30–32, 49–50
  - kernel and, 15
  - kernel upgrades using RPMs, 216
  - LinuxLoader (LILO) in, 14, 16–21
  - logical partitions (dev/hda) in, 12–13, 23–26, 31
  - logical structures created in, 11–34
  - Master Boot Record (MBR) in, 14, 18, 20
  - master slave configuration in, 31–32
  - memory and, 8
  - mount point in, 8–9, 24
  - multiple drive configurations and, 30
  - multiple swap partitions in, 29
  - operating systems, multiple, 13
  - paging of memory in, 8–9
  - Partition Magic for, 12, 16
  - partition managers for, 13
  - partition table in, 13–14
  - primary partition in, 12, 23
  - proc/ide directory in, 31
  - RAID, 33–34, 74–83
  - random access memory (RAM) in, 8
  - Red Hat service agreement and, 13
  - root file system and, 15, 21–22
  - SCSI drives and, 23, 32–33, 71–83
  - separate partitions for directories, pros and cons, 27–30
  - size of partitions in, 12–13
  - sizing swap space in, 9–11
  - swap files in, 7–9, 25–26
  - swap partition in, 7–9, 21–22, 25–26, 80

- swap space options in, 7–11
- system partitions in, 26–34
- thrashing and, 8
- user partitions in, 26–34
- usr directory, 29
- var directory, 29
- Web server, 80
- partition table, 13–14
- password, root password selection, 56–57, 88
- PCI access mode option, in kernel configuration, 236
- PCI quirks option, in kernel configuration, 236
- PCI support option, in kernel configuration, 236
- PCMCIA installation, 36
- peripheral component interconnect (PCI), in video configuration, 126
- permissions, in file sharing, NFS, 117–119
- phosphor screen, in displays and monitors, 133, 145
- pixels, 127
- Plug-and-Play support, 2
  - in boot process, 15
  - in kernel configuration, 237
- ports, 3
- power on self test (POST), 15
- preinstallation inventory of system, 1–6
  - boot disks and, 4
  - disk operating system (DOS) in, 4
  - existing equipment in, 3–4
  - memory requirements in, 4
  - recently purchased equipment in, 3
  - sample list of components in, 5
  - system identification in, 2–4
- primary partition, 12, 23
- print queues, 185–186
- printer capability database (printcap), 173–177
- printer configuration, 91, 173–194
  - commands and utilities for, basic, 173–177
  - default printer in, 183
  - IDE system installation, 59
  - line printer (lpr) command in, 173–177, 181–185
  - line printer spooler daemon (lpd)
    - command in, 173–177, 180–186, 190
  - local printing configuration for, 177–186
  - LPRng package for, 174–176
  - print queues in, 175–178, 185–186
    - definition for, 178
    - type of, 178
  - printer capability database (printcap) in, 173–177
  - printer creation in, 177–178
  - printer driver for, 179–180, 183, 189–190
  - printtool utility for, 173, 176–177, 181–186
  - remote printing, 186–194
    - destination (HostA queue) for, 192–194
    - etc/hosts.lpd file in, 190–192
    - hostC queue for, 188
    - line printer spooler daemon (lpd)
      - command, 190
    - prerequisites for, 186–194
    - printer driver for, 189–190
    - queue type defined for, 189
    - TCP/IP and, 186–188
    - verifying printing in, 192–194
  - remove entry from print queue (lprm)
    - command in, 185, 186
  - spool directory for, 185–186
  - technical details of, 185–186
  - test page printing, 184
  - testing print commands for, 181–185
  - testing the printer for, 181
  - view queue entries (lpq) command in, 182, 184
- printtool utility, 173, 176–177, 181–186
- proc/ide directory, 31
- proc/scsi/scsi, 32
- processor types and features, in kernel configuration, 232
- public domain status of Linux, 36
- queues, print, 175–178, 185–186
- RAID, 2
  - partitioning for, 33–34, 74–83
  - in SCSI system installation, 68, 74–83
- RAM disk
  - in source tree kernel install, 270–271
  - support option, in kernel configuration, 240
- RAMDAC, in video configuration, 128, 130
- random access memory, video (VRAM), 126–128, 162
- random access memory (RAM), 8–11
  - in boot process, 15
- read only memory (ROM), 15
- README.Config file, in video configuration, 166
- reconfiguring the system, 101
- Red Hat service agreement, 13
- Redundant Array of Inexpensive Disks (See RAID)
- refresh rate, video, 130, 136, 144, 159
- remote printing (See printer configuration, remote)
- remove entry from print queue (lprm)
  - command, 185–186

- rescue mode, for IDE system installation, 45–46
- resolution, video, 127, 130, 145–147, 160
- rmmod module, source tree kernel install, 251, 253
- root file system
  - in boot process, 15
  - partitioning, 21–22
- root password
  - IDE system installation, 56–57
  - SCSI system installation, 88
- RPM packages, 195–212
  - applying maintenance using, 200–205
  - backward compatibility among, 196
  - confirming upgrade in, 205
  - contents and function of, 197–200
  - current version confirmation in, 203
  - database and binary files for, 196
  - definition of, 196
  - dependencies among, 196
  - determining which package to upgrade using, 202–203
  - directory for, 200
  - FTP update files for, 200–202
  - hash (h option) of, 204
  - installation (i option) of, 204
  - installation of, 196
  - kernel upgrade using (*See* kernel upgrades using RPMs)
  - man page for, 196–197
  - naming conventions for, 197
  - NFS installation, 103
  - procedures for upgrading, 204–205
  - query options for, in rpm utility, 198
  - rpm utility for, 196
  - SRPM vs., 197
  - super daemons and, 202
  - updating, 195, 200–205
  - verify (v option) of, 204
  - XF86config file for, 195
  - XF86Setup installation and, 195, 209–212
  - XFree86 package and, replacing and upgrading, 195, 205–209
  - Xinetd for, 202–203
- SCSI emulation support option, in kernel configuration, 239
- SCSI low level drivers, in kernel configuration, 245–247
- SCSI support, in kernel configuration, 244–247
- SCSI system installation, 68–101
  - authentication configuration in, 88, 91
  - boot message for, 70
  - custom, 70
  - development packages in, 91
  - DOS/Windows connectivity in, 91
  - firewall configuration in, 86
  - GNOME in, 92
  - IP addresses in, 85–86
  - KDE in, 91
  - kernel development package in, 91
  - keyboard specification in, 70
  - language selection in, 70
  - language support in, 88
  - linuxconf in, 92
  - LinuxLoader (LILO) installation/  
configuration in, 83–85
  - Master Boot Record (MBR) install in, 50–51
  - monitor setup in, 95–96
  - mouse selection in, 86–88
  - multimedia support in, 91
  - network configuration in, 85–86
  - networked workstations in, 91
  - NFS Server in, 91
  - non-root user addition in, 88
  - package group selection in, 91–92
  - partitioning and, 23, 32–33, 71–83
  - printer in, 91
  - proc/scsi/scsi in, 32
  - RAID array in, 2, 68, 74–83
  - root password selection in, 88
  - screen and installation of selected items in, 93–94
  - time zone configuration in, 88
  - troubleshooting, 99–100
  - user interface/X in, 92
  - utilities in, 92
  - video configuration in, 92–93, 96–98
  - Web server in, 91
  - Welcome screen in, 70
  - X Window System in, 91, 98–99
  - XFree86-xf86cfg in, 92
- security
  - kernel upgrades using RPMs, 214–215
  - NFS installation, 108–109
- server, NFS, 105–109
- Session Message Block (SMB) installation, 36
- shadow mask, 133, 145
- signal generation, video, 125–132
- Simple Mail Transfer Protocol (SMTP),  
memory requirements, 10
- software requirements, in source tree kernel install, 263–265
- sound cards, 3
- source code, for source tree kernel install, in tar file format, 258–259

- source tree (*See* kernel install using source tree)
- spool directory, 185–186
- SRPM files, RPM packages vs., 197
- standard build, 103
- sunrpc module, in source tree kernel install, 253
- super daemons, RPM packages, 202
- SuperProbe for video information, 151–154
- swap files, 7–9, 25–26
- swap partition, 7–9, 21–22, 25–26, 80
  - multiple, 29
- swap space options, 7–11
- synchronization, video, 134–136, 144, 159–160
- system partitions, 26–34
- System V IPC option, in kernel configuration, 236
- system.map files, in source tree kernel install, 269–270
- TCP/IP
  - NFS installation, 100
  - remote printing, 186–188
- TCP/IP networking option, in kernel configuration, 241
- Telnet, IDE system installation, 57, 59
- temporary mount, 116
- text menu installation vs. GUI, 35–36
- thrashing, 8
- time zone
  - CMOS and, 55–56, 88
  - IDE system installation, 55–56
  - Network Time Protocol (NTP), 56
  - SCSI system installation, 88
- timing, in file sharing, NFS, 117
- tmp directory, 27
- troubleshooting, 99–100
- unconfigured file, 101
- UNIX, boot disk creation, 36, 38–40, 64, 94–95
- UNIX domain sockets option, in kernel configuration, 241
- user ids (uids), in file sharing, NFS and, 112–113, 118–119
- user interface/X, 92
- user management
  - IDE system installation, adding non-root users, 57–58
  - SCSI system installation, adding non-root users, 88
- usr directory/partitions, 26–34
- usr/sbin/sys-unconfig file, 101
- utilities, 59, 92
- var directory, 27, 29
- vertical scan frequency (VSF), 136, 144
- vertical synchronization frequency (VSF), 130
- vertical synchronization pulse (VSP), 135–139, 144, 159–160
- VGA memory clock, 132, 141–143
- video and video configuration, 125–171
  - accelerator for, 128
  - adapter hardware in, 157
  - automatic boot from configuration, 165
  - bandwidth in, 141–143
  - BIOS and, 156
  - buses and slots for, 126, 128–132
  - cathode ray tube (CRT) displays for, 126
  - clock, dot clock or VGA memory clock in, 132, 141–143
  - color depth in, 160
  - custom monitor setup in, 158–159
  - desktop installation in, 168–171
  - digital to analog converters (DACs) in, 132
  - displays and monitors for, 132–139
  - dot pitch in, 144–147
  - factors affecting performance in, 140
  - feedback on configuration in, using SuperProbe, 161
  - GNOME systems and, 168
  - horizontal frequency in, 144
  - horizontal synchronization pulse (HSP), 135–139, 159–160
  - IDE system installation, 61–62, 65–67
  - image data transfer rate in, 130–131
  - interlacing in 147–149
  - KDE systems and, 168
  - mode, video mode in, 127–129, 163–164
  - modeline numbers and codes for, 140
  - monitor age vs. performance of, 141
  - monitor type in, 157
  - NFS installation, 100–101
  - performance and capability of monitors in, 139–147
  - pixels in, 127
  - preparation for, 149–167
  - probing for hardware in, 156
  - problems and troubleshooting in, 166–167
  - RAMDAC or chipsets for, 128, 130, 156
  - README.Config file and, 166
  - refresh rate in, 130, 136, 144, 159
  - resolution in, 127, 130, 145–147, 160
  - sample configuration in, 154–167
  - SCSI system installation, 92–93, 96–98
  - signal generation in, 125–132
  - SuperProbe for information on, 151–154
  - synchronization in, 134–136, 144, 159–160



- testing configuration of, 164–165
- text message test, 165
- unsuccessful Xconfigurator probe and, 161–163
- vertical synchronization frequency (VSF) in, 130, 144
- vertical synchronization pulse (VSP) in, 144, 159–160
- video adapter memory (VRAM) in, 126–128, 162
- Video Electronics Standards Association (VESA) standards in, 130
- X package installation for, 167
- X server setup for, 167
- X Window System scratch installation in, 167–171
- Xconfigurator for, 125, 127–128, 153–156, 168, 195, 209
- Xconfigurator problems and troubleshooting for, 166–167
- XF86Config file for, 125, 139, 150–151, 155–156, 166
- XF86Setup for, 195, 209–212
- XFree86 in, 150–151, 205–209
- video card, 3, 44
- video chips, 3
- Video Electronics Standards Association (VESA) standards, 130
- view queue entries (lpq) command, 182, 184
- Web server, 80, 91
- Windows, file sharing with Linux system, 119–123
- X server
  - IDE system installation, 48
  - in video configuration, 167
  - XF86Config file for, 150–151
  - XFree86 in, 150–151
- X Window System, 4, 9, 67–68, 91
  - IDE system installation, 59
  - SCSI system installation, 98–99
  - in video configuration, scratch installation in, 167–171
  - XF86Config file for, 150–151
- xconfig files, 229–230
- Xconfigurator (*See also* video and video configuration), 125, 127–128, 153–155, 168, 195, 209
  - problems and troubleshooting for, 166–167
- XF86Config file, 125, 139, 150–151, 155–156, 195
  - Xconfigurator and, 153–154
- XF86Setup, 209–212
  - confirming update of, 212
  - directory for, 211
  - FTP files for upgrading, 211–212
  - obtaining update packages for, 211
  - tutorial for, 210–212
  - update vs. install files for, determining, 212
  - video and, 195
- XFree86, 92, 150–151
  - confirming upgrade of, 209
  - obtaining necessary upgrade files for, 207
  - replacing and upgrading, 205–209
  - RPM files to replace in, identity of, 206–207
  - RPM packages, 195
  - upgrade vs. install packages in, determining, 208–209
- Xinetd, 202–203
- Xlink Technology, 119–123
- zlilo utility, 249